Issue 12 | Sep 2007 | Blenderart Magazine

blender art MAGAZINE

Blender learning made easy

Texturing special!

Multi Layered Plastic Shaders

Tangent Space Normal Maps

Creating Realistic Flame

Blender to Kerkythea

Texture Seam Removal

Roof Tops - Making Of

Coverart - Cristian Mihaescu - 'Eggland'

blender art MAGAZINE

EDITOR/DESIGNER
Gaurav Nawani gaurav@blenderart.org

MANAGING EDITOR
Sandra Gilbert
sandra@blenderart.org

WEBSITE
Nam Pham
nam@blenderart.org

PROOFER
Phillip A. Ryals
Kevin C. Braun
Derek Marsh

WRITERS
Abhihsek Gupta
Adam Friesen
Alex (blenditall)
Andi Schumann
Claas Kuhnen
Eric Pranausk
Krzysztol Zwolinski
Olivier Saraja

COVER ART
Cristian Mihaescu - 'Eggland'

Sandra Gilbert
Managing Editor

One of the first skills you will need to learn is how to observe!

You know that you have finally become a real “blenderhead”, when not only do you look at objects around you and break them down into their various vertices, edges and faces, but when you become awestruck by the infinite variety of textures in the world around you. You will be walking along, minding your own business when it will grab you, the way rust is peeling on a signpost, the way a flower petal looks with morning dew on it. You will notice how the light affects the texture and next thing you know, you have gone from awestruck to breaking it down and figuring out how to recreate it in blender.

Welcome to the wonderful world of Texturing. In this issue we will be taking a look at how to create great textures to enhance your images and animations. Creating great textures is a skill like any other and one that can be learned through practice and experimentation. One of the first skills you will need to learn is how to observe everyday textures. What makes them unique, what makes them “work”. Then with that knowledge and a good understanding of Blender's texture tools, you are on your way.

In this issue, not only are we covering how Blender's texture tools work, but how to effectively use them to create truly great textures.

We have gathered up some great tutorials that will show you how to remove UV Seams and create Normal Maps. Then you can try testing your new skills by following along to create a great Skin Shader and realistic plastic shaders. Which oddly enough, is not as easy as one would think and yet not as scary or difficult as the last statement would imply. In 'Keeping the Flame', we see how to set up and create a realistic flame. Then we go behind the scenes with a look at how 'Rooftops' was created, while in 'Tangrams of Light' we learn how lights affect textures and how the whole can be used to create just the right mood.

So go grab a cup of coffee and curl up for some great, informative reading, sure to change the way you approach your next texturing adventure.

Happy Blending!
sandra@blenderart.org

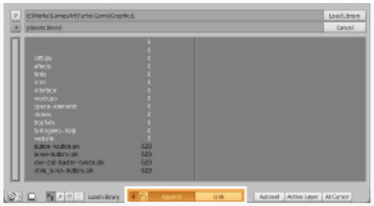
A look at various tools and Texturing techniques to enhance your images.

At some point in your blender exploration, it will happen. The perfect material/texture for your current masterpiece is one that you have already created in a previous project or one that has been created by a generous community member that has offered to share his/her materials with the community. So how do you go about getting that perfect material/texture into your current project? Well it couldn't be easier.

Blender has two options for reusing materials, Linking and Appending *(actually both these options can be used for all of blender's assets. Blender assets can include actions, armatures, cameras, images, IPOs, lamps, materials, meshes, objects, scenes, text, textures, and world, etc.).*

Appending will place a independent copy of the material into your new project.

Linking to a material will link to the original file that contains the material, meaning that any changes to the original file will also be saved to the new file that you linked it to. This is a useful option when working on a large project that will generate a large number of files that may be worked on by one or more people. This will allow any changes made during production to be propagated throughout all needed files without the added work of updating each file individually.



Okay, on to actually seeing how to use these options. Both options are started the same way:

1. From your new/current project file, either by going to File› Append or Link or using the hotkeys Shift + F1, open the file browser window. (All blend files can be used for Appending or Linking.)
2. Browse to where you have saved the blend file that contains your perfect material.
3. Click on the file name of the desired blend file which will open the library list of Appendable/Linkable assets.
4. In this case we are looking for materials, so click on Material.
5. That will open a list of all available materials in that blend file.
6. At the bottom of the file browser there are two buttons, one for Append and one for Link. Push whichever option you have chosen for your project.
7. Right Mouse Click on the Material you want to Link or Append, to highlight it and then Middle Mouse click to confirm (load) your material to your new project file.
8. Now your material will be available to assign to whatever object/model you wanted it for.

There, just as I promised, easy as can be. With practice, you won't even have to think about it, you will just click your way to reusing your blender assets.

Bump Mapping

What is a Bump Map? Bump maps are textures that store an intensity, the relative height of pixels from the viewpoint of the camera. The pixels seem to be moved by the required distance in the direction of the face normals. You may either use greyscale pictures or the intensity values of a RGB-Texture (including images) (definition taken from the blenderwiki).

Well that is a nice definition and all, but what does it mean to the average artist? It means we can fake details and geometry that would be time consuming to create or result in a model that due to high vertex/face counts would take entirely too long to render (that of course is for those of us that are impatient with long render times.)

Let's hear it for faking! Now let's look at how to use Bump maps.

1. Add a new Material in the Material Buttons, assign color of choice
2. Click on the Texture button (F6)
3. Click in a Blank texture slot
4. Choose an texture type (there are many to choose from, some procedural textures don't work as well as others, for more information on the different texture types and their uses you can refer to the blenderwiki:
5. Apply the Texture in the Material buttons Map To panel by clicking the Nor option. The strength of the effect is controlled with the NumButton Nor on the same panel.

Multiple Bumps maps can be used together to create any number of detailing effects. UV map layouts can be used in combination with procedural textures to create more complex and realistic effects.

Render Baking

Render baking creates 2d bitmap images of a mesh object's rendered surface. These images can be re-mapped onto the object using the object's UV coordinates. Baking is done for each individual mesh, and can only be done if that mesh has been unwrapped. While it takes time to set up and perform, it saves render time. If you are rendering a long animation, the time spent baking can be much less than time spent rendering out each frame of a long animation.

Use Render Bake in intensive light/shadow solutions, such as AO or soft shadows from area lights. If you bake AO for the main objects, you will not have to enable it for the full render, saving render time.

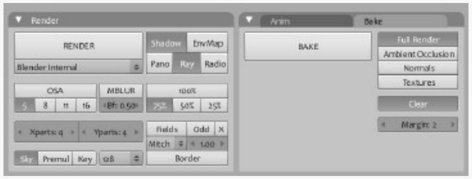
Use Full Render or Textures to create an image texture; baked procedural textures can be used as a starting point for further texture painting. Use Normals to make a low-resolution mesh look like a high-resolution mesh. To do that, unwrap a high-resolution, finely sculpted mesh and bake its normals. Save that normal map, and Map To the UV of a similarly unwrapped low-resolution mesh. The low-resolution mesh will look just like the high-resolution, but will have much fewer faces/polygons.

Advantages:

- Can significantly reduce render times
- Texture painting made easier
- Reduced polygon count
- Repeated renders are made faster, multiplying the time savings

Disadvantages:

- Object must be UV-unwrapped.
- If shadows are baked, lights and object cannot move with respect to each other.
- Large textures (eg 4096x4096) can be memory intensive, and be just as slow as the rendered solution.
- Human (labor) time must be spent unwrapping and baking and saving files and applying the textures to a channel.



Options

Full Render: Bakes all materials, textures, and lighting except specularity and SSS.
Ambient Occlusion: Bakes ambient occlusion as specified in the World panels (F8). Ignores all lights in the scene.
Normals: Bakes camera-space normals to an RGB image.
Textures: Bakes colors of materials and textures only, without shading.
Clear: If selected, clears the image to selected background color (default is black) before baking render.
Margin: Baked result is extended this many pixels beyond the border of each UV "island," to soften seams in the texture

Process

Windows Users do AO first: If you are running Blender on a Windows operating system, you may have to first bake Ambient Occlusion before baking any other option. If you do not bake AO first, you may get the error message "No Image to Bake To" and will not be able to bake anything for that mesh.

1 In a 3D View window, select a mesh and enter UV/Face Select mode
2 Unwrap the mesh object
3 In a UV/Image Editor window, either create a new Image or open an existing Image. If your 3D view is in Textured display mode, you should now see the image mapped to your mesh. Ensure that all faces are selected.
4 With your mouse cursor in 3D View, press Ctrl Alt B to popup the menu of available baking choices. Alternatively, access the Bake panel in the Buttons window, Scene (F10) context, Render sub-context.
5 Bake your desired type of image: Full Render, Ambient Occlusion, Normals, or shadeless Textures.
6 After computation, Blender replaces the image with the Baked image.
7 Save the image in the UV/Image Editor window via Image->Save

(Render Baking information taken from the blenderwiki) Further tips on usage can be found at the blenderwiki

Texture Painting in Blenderart

Blender features a built-in paint mode, called Texture Paint, designed specifically to help you edit your UV Textures and Images quickly and easily in either the UV/Image Editor window or the 3D View window.

In the 3D window in Texture Paint mode, you paint directly on the mesh. In the UV/Image Editor window, you paint on a flat canvas that is wrapped around the mesh using UV coordinates. Any changes made in the UV/Image Editor win-

-dow shows up immediately in the 3D window, and vice-versa.

A full complement of brushes and colors can be selected from a floating Image Paint panel in the UV/Image Editor, or a Paint panel in the Buttons window, Editing (F9) context. Brush changes made in either panel are immediately reflected in the other panel.

When satisfied or at intermittent intervals, please save your image using the UV/Image Editor window.

Once you have un-wrapped your model to a UV Texture (as explained in previous pages), you have to:

- Load an image into the UV/Image Editor (Image->Open->select file) , or
- Create a new image (Image->New->specify size) and saved it to a file (Image->Save->specify file).

You cannot paint on a mesh in Texture Paint mode without first unwrapping your mesh, and doing one of the above steps. After you have done these two things, you can modify the image using the Texture Paint mode. Once you do:

- In the 3D View window, select Texture Paint mode from the mode selector in the window header, and you can paint directly onto the mesh.
- In the UV/Image Editor window, enable Texture Painting in the Image menu (shown to the right).
- In the UV/Image Editor header, click the magic pencil (highlighted to the right).

At this time, you may choose to show the alpha (transparency) channel by clicking the button to the right of the magic pencil. The dot icon to the right of that button allows you to paint the alpha channel by itself.

Once you enable Texture Painting, your mouse becomes a brush. To work with the UV layout (for example, to move coordinates) you must disable Texture Painting. To work with the mesh in the 3D View (for example, to move it in 3D space), or select other objects, you must leave Texture Paint mode.

When you enable Texture Painting, use the View->Paint Tool option in the UV/Image Editor window or the Paint panel in the Buttons window to modify paint settings.

All painting you perform in either window will be instantly reflected in the other window (if the 3D View is in textured viewport mode). However, the modified texture will not be saved until you explicitly do so by Image->Save in the UV/Image Editor window.

See Outline: *If you want to paint directly on the mesh in the 3D View window, change to UV/Face Select Mode first to show the edge outline of the mesh, then switch to Texture paint mode, which visually overlays the previous mode, but keeps the outline.*

As soon as you enable Texture Painting or switch to Texture Paint mode, a Paint Panel becomes available in the Editing (F9) buttons. This panel has all the same controls as those available in the Paint Tool. Use this panel if you are only working in 3D view in order to change brushes (colors, patterns, function). (Texture Painting information taken from the blenderwiki)

Further tips on usage can be found at the blenderwiki blenderwiki

This is a short overview of some of the available blender texture tools that you may or may not have been aware of. For more information on texturing tools and techniques (as well as other areas of blender) refer to the blenderwiki documentation.

A wealth of information is available there for you to browse through and use. A big thanks goes out to our documentation team for keeping our blenderwiki up to date and easy to use.

Image1: Work examples from Phillip Carrizzi jewelry work

Multi Layered Plastic Shaders

- by Claas Kuhnen

Introduction

Whenever you want to create a digital material select the best objects with that surface. Images can help as well, but only with the physical object in your hand is it possible for you to study the interaction of light and surface in detail. And 45 degrees above the X-axis. Select the vertex to extrude [RMB] snap the cursor to the vertex [Shift-S] Constraining moves to an axis and dimensional input works for extrusion [E] rotation [R] move (grab) [G] and scale [S]. However with scale you will need to use a scale factor rather than a dimensional input. This is easily calculated by dividing the required finished size by the existing size of the object.

1. Basic Setup:

The majority of shaders we create and the way they are calculated are in general comparable to an egg shell. The light that hits the surface, gets reflected, it does not penetrate it and illuminate the inside. This limitation for example also formed the expression plastic rendering.

The current implementation of Sub Surface Scattering tries to simulate the penetration of light into the volume of the model to create proper mass illumination, enabling Marble, wax, skin, and other materials that can be simulated in a more realistic way. However not every plastic surface is the same. And as simple as it sounds, good plastic shaders can also take quite a bit of work to get right. Specifically when the surface finish is layered, a lot of extra steps have to be included to produce a realistic shader. The material setup should be similar to the real world counterpart.

These type of surface finishes very often have a primer layer, two effect layers with iridescent particles, and one clear lacquered finish layer. The most common and visible example is in transportation design and they are also commonly found in both product design and jewelry design. See title image for reference, you can se more examples of Phillip Carrizzi here.

As you can see in the example, we have different specular reflections, we have true mirror reflections, and different colors. We have to analyse this structure first to be able to translate it into a shader. If we break those visual elements down we would get the following hierarchy:

Finish layer:
Translucent, high polished, with a strong burning specular reflection / highlight and a mirror effect with a slight Fresnel value.

Second layer:
Translucent, with iridescent particles in a specific color, producing an individual specular reflections pattern.

Third layer:
Translucent, with iridescent particles in a specific color, producing an individual specular reflection pattern.

Primer layer:
Opaque, basic color for the main object.

In addition to the layering, we also have to look at the scene and watch for indirect and direct lighting differences. For example, when you watch cars with iridescent shaders drive into a shadow, you will notice that the car lacks changes. Because of a lack of direct light, the specular highlights will change, the diffuse pattern will change.

When you take a look at a car very closely, you will also notice that the specular reflection is not just plain white. It has a nice set of rainbow colors. This, for example, is true with many plastic materials when they are in direct light. But it is all dependent on the angle of light impact and from where you see the object. Those colored specular reflections are very often an overlooked characteristic. For example, you also find them in brushed metal and coarsely brushed aluminum parts.

So how do we translate this into a digital shader? Lets use the material mixer. Technically we do not need the same layer hierarchy. The primer layer, for example, could be combined together with the mirror reflection and burning highlight reflection. The reason why I would select the material mixer is simply because of the flexibility to create individual materials and the option to mix only diffuse and specular together.

In Blender this material hierarchy could look like the following:

Main base shader:
Diffuse color value
Specular reflection - hard burning
Mirror reflection value - very low Fresnel set up
Specular light model : WardIso

First specular shader:
No Diffuse output
Specular reflection - focused spread
Color Specular - Color value set to 0.02
Specular texture is scaled down: Size 0.01
Specular light model : Blinn

Second specular shader:
No Diffuse output
Specular reflection - wide spread
Color Specular - Color value set to 0.02
Specular texture is scaled down to: Size 0.02
Specular light model : Blinn

But how do we now create the pointy highlight texture? As a specular texture, we can use the Voronoi with the Col1 tab selected. This texture can be used for the bump map channel as well as the color specular channel.

The bump map will produce the grainy look, the Color specular will give the specular reflection some color. Because we do not use any diffuse, the specular will still be spread out by the texture size and normal value.

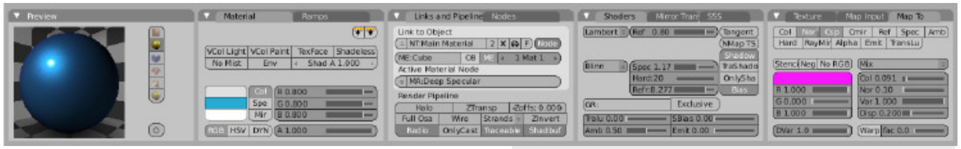

Preview | Material | Ramps | Links and Pipeline | Nodes | Shaders | Mirror Tran | SSS | Texture | Map Input | Map To
VCol Light | VCol Paint | TexFace | Shadeless
No Mist | Env | Shad A 1.000
Col | Spe | Mir
R 0.800
G 0.800
B 0.800
RGB | HSV | DYN
A 1.000
Link to Object
NT:Main Material | 2 | F | Node
ME:Cube | OB | ME | 3 Mat 1
Active Material Node
MA:Deep Specular
Render Pipeline
Halo | ZTransp | Zoffs: 0.000
Full Osa | Wire | Strands | ZInvert
Radio | OnlyCast | Traceable | Shadbuf
Lambert | Ref 0.80 | Tangent | NMap TS
Blinn | Spec 1.17 | Hard:20 | Refr:8.277
Shadow | TraShadow | OnlySha | Bias
GR: | Exclusive
Tralu 0.00 | SBias 0.00
Amb 0.50 | Emit 0.00
Col | Nor | Csp | Cmir | Ref | Spec | Amb
Hard | RayMir | Alpha | Emit | TransLu
Stencil | Neg | No RGB | Mix
R 1.000
G 0.000
B 1.000
DVar 1.0
Col 0.091
Nor 0.30
Var 1.000
Disp 0.200
Warp | fac 0.0

Image2: Blender First Specular Layer Material

WardIso will produce very sharp, hard edged reflections which are a good indicator of a well polished surface. Blinn, which has a little less burn quality, can be used for the iridescent particles of the following two inner layers. The specular reflections of those two layers are less strong compared to the top layer.

By setting the Spec value inside the Material Button from white to a different color we can also tint the complete specular sparkles into a desired color.

As you can see in the reference objects, the two layers have two different colors.

Inside the material node mixer, we combine these three materials together.

The two specular reflection materials are being added together and their color result is being added together with the main shader material.

Add and Mix produces different results. In our real world objects they add together. With Mix you can specify if one input dominates more. Add defines how much the second channel is being added to the first one. Here we also deactivate the Diffuse output of the two Specular Reflection materials.

This setup should produce a rough representation of the desired plastic object.

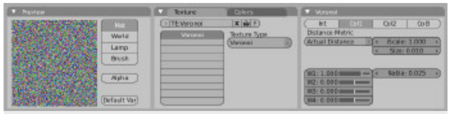

Preview | Texture | Colors | Voronoi
Mat | World | Lamp | Brush
Alpha
Default Var
TE:Voronoi
Voronoi
Texture Type
Voronoi
Int | Col1 | Col2 | Col3
Distance Metric:
Actual Distance
Scale: 1.000
Size: 0.010
W1: 1.000
W2: 0.000
W3: 0.000
W4: 0.000
Nabla: 0.025

Image3: Blender First Specular Layer Texture

2. Refined Settings:

Now it is time that we setup all material values correctly to our needs. Using the material mixer, we can always quickly turn on/off complete networks,
a specific color output, or change the mixing proportions.

Looking at the reference image we can identify that the first specular layer is very fine, is very close to the main burning reflection, and also shows a similar while less strong burn intensity. This tells us that the first specular layer has to be fine and not spread too widely. It should start to blend visually together with the main reflection and not spread too much over the main body.

The reason for that is that the particles in the second layer seem to be denser together and thus react more to the incoming light. This also means the light direction is important to keep in mind.

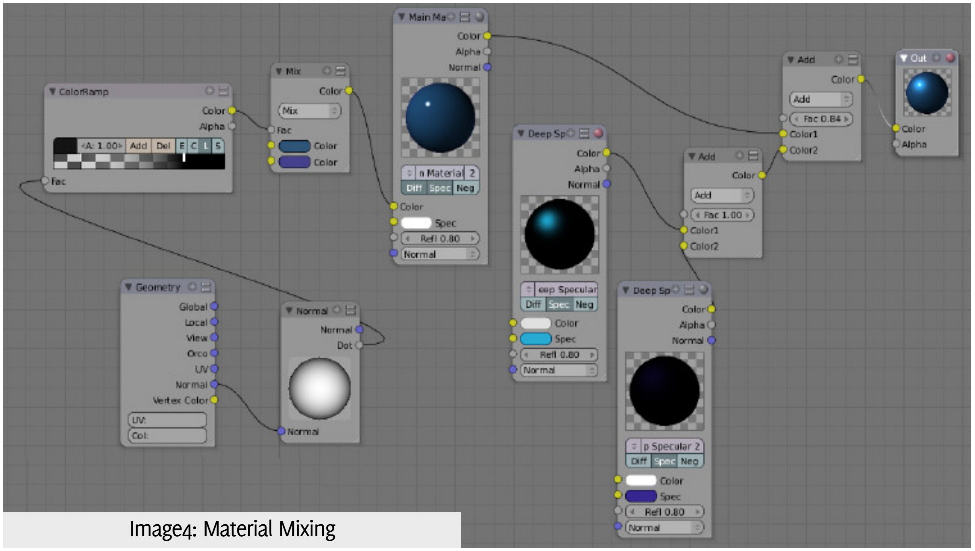
In my example I set the Spec value to 1.17, Hard to 20, and Refr to 8.27. This produced the desired shape and coverage. With the Nor value we can shape the material in a way that the individual reflections will stand out or blend more together. I want them to blend a little more together, so I use a value of Nor 0.1.

The next specular layer, the first layer applied to the body has a much coarser pattern. It also spreads much more over the entire body and has a much less strong intensity. I set Spec to 0.52, Refr to 3.47, and to get a more spread Hard to 10.

3. Comparison:

This produces a quite close representation. Depending on scale and view distance, we have to adjust the texture values. However if we compare it to the real object, we also notice a significant difference. The spread in the digital model is too even. To help this it would make sense to actually paint the textures in Photoshop instead of using a procedural texture. We would only need a coarse and a fine point map.

The transparent areas in between will allow the surrounding area to remain transparent while keeping the point size the same. Something difficult to create when you can only use scale to bring points closer or more far away from each other.



A: 1.00

Refl 0.80

Refl 0.80

Refl 0.80

Fac 1.00

Fac 0.84

Image4: Material Mixing

Additionally, it will be very visually interesting to also add a soft iridescent color change over the complete visible body, mainly at the object edges. Make the color changes very soft. However if we look at Phill's object, we can see that there is no typical iridescent effect. The edges are darker simply because of the lack of glowing metal pigments. We plainly see the base color of the primed surface.

Also keep in mind when you work from an image, that the reflection in your image is showing the environment of that shot. Of course, in that case it will be difficult to produce the same looking reflection in your work.

Reflections always depend on what you put your object into in terms of physical objects around it and the light setup. If nothing is there to reflect, then your reflections will also look logically empty, which can be the case in studio setups.

4. Indirect Illumination Trick:

In addition, because in nature there are no 100% dark shadows and every illuminated surface illuminates the environment, we also have to take this indirect illumination into account. A very simple way to make our model look better without using time consuming AO, is just to utilize a filler lamp. Next to the speed advantage, this also has the ability to simulate color bleeding, which Blender's AO cannot produce at this point.

We can easily simulate the color bleeding of the ground color to the object, by giving the light the same color as the ground plane. We just need to place a lamp below the ground mesh and let it shine upwards.

But which lamp to use? Sun produces perfect linear light rays, but we cannot specify an endpoint. Point and spot are emitting light from one point. With the fall off value, we could specify where the reflected light would stop. Only the lower part of the model would receive bounced light. But we would need to create an array of lamps to produce an even illumination.

Area lights produce a nice even illumination and with distance, we can somewhat control the fall off a little bit. Let's use the last one. With a very low Dist value of 0.3, I place the lamp slightly below the ground. The more I move it away, the more the light intensity is lowered. Please keep in mind that in case you want to increase the area of the lamp use the Size slider and do NOT scale the area lamp. If you scale, you also scale up the distance of the fall off.

In case you change the Size value, you increase the surface from which light will be emitted and thus the overall illumination intensity will be different. Scale it down and the light rays will be compressed and illumination intensity will be very strong. Use the Distance indicator as a reference for scaling. See image5.

First, I just set a distance and then scaled the lamp to the right fall off result. In the following step, I did use Size and Energy to get the desired area illuminated with the desired strength. Scale and Blender dimensions are the key here.

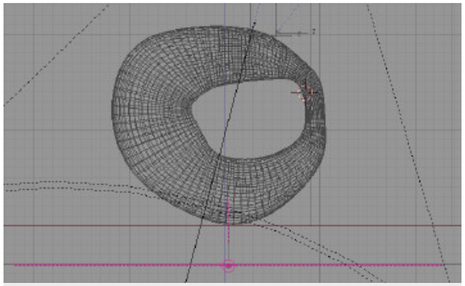


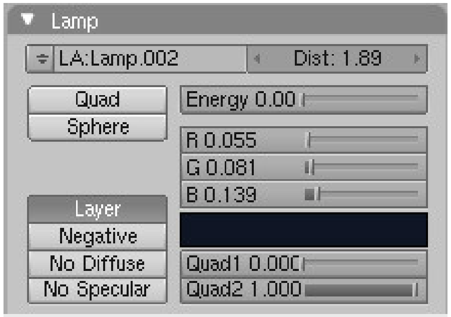
Image5: Area Light.

Inside the Blender scene are two cameras. One is nearly parallel to the ground. I use that one to check how well the area light is illuminating the mesh from below. I use multiple cameras to quickly switch between different views for testing light and material settings. I prefer having individual cameras, than having to move the camera all the time.

In addition we could also simulate the color bleeding a little bit here. All we need to do is to create a spot light with a very soft edge and let it shine downwards onto the floor.
We have to lower the light intensity and also give it a color equal / darker than the body color of the plastic. We want to limit the Lamp to only illuminate the ground plane.

For that we have to put the ground plane mesh and the lamp onto the same layer and activate the layer option for the lamp under the lamp buttons.

The illumination should only be a very small effect. Depending on ground and body color the bleeding effect would be more or less visible.



Lamp

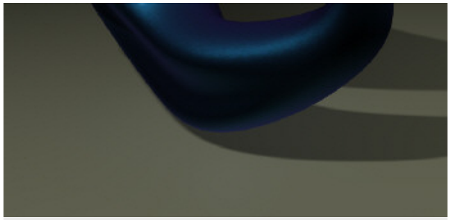
LA:Lamp.002 Dist: 1.89

Quad
Sphere
Layer
Negative
No Diffuse
No Specular

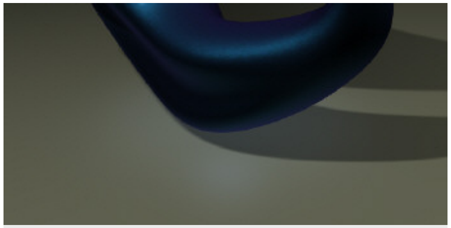
Energy 0.00
R 0.055
G 0.081
B 0.139
Quad1 0.00C
Quad2 1.000

Image6: Spot Light Layer.

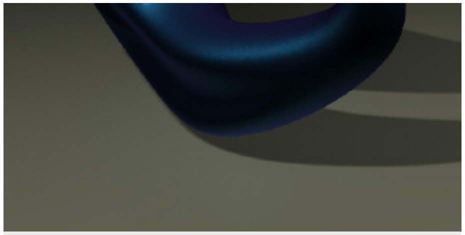
Spot Light Intensity Variations.



None: Energy 0



Too Much: Energy 1.00



Just Right: Energy 0.50

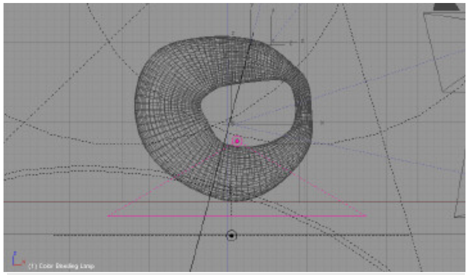


Image 6: Spot Light.

This should give you a good idea about what to look out for. The supplied test scene for example, would require some work in the light set up.

Without a scene around it, all reflections look dark. It even looks like there are no reflections on the object.

All this will change with some more work and actually having the bracelet relating to an object. Imagine a scene in which the bracelet is on an actual arm. With the reflection of the skin and the environment, it will look very different.

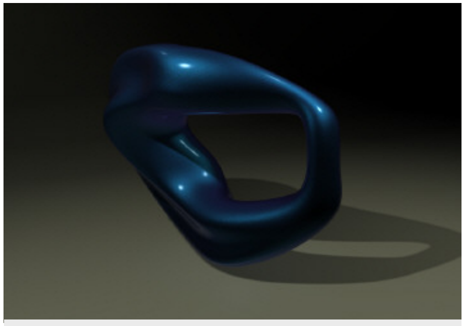


Image 7: Finale Rendering with slide Fresnel reflec-

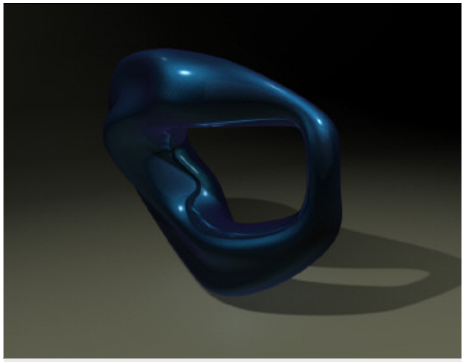


Image 7: Finale Rendering with slide Fresnel reflection value .

Note: the distraction inner reflection, but the nice window like reflection on the upper and right part.

Claas Eicke Kuhnen
Germany

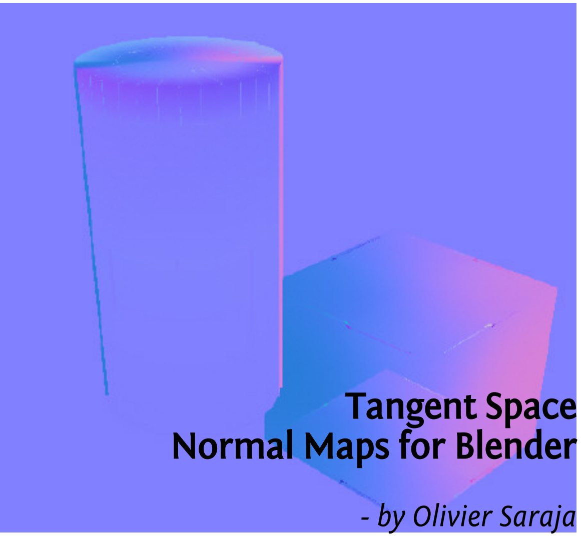
MFA 3D Studio Jewelry/Metal
Bowling Green State University, USA
Focus in Functional Metal Art and 3D Digital Art

Dipl. Des. (Fh) Color – Advanced Color Concepts HAWK University of Applied Science and Art, Germany Focus on Functional Graphic and Product Design

After grad school I taught for one year at the University Wisconsin-Stout where I introduced Blender for industrial design and interior design to students. Through that exposure to the students I focused more on researching the usability of Blender for this field.

It has increased my knowledge and understanding to see how NURBS and SDS can be combined in a professional work flow for CAD and Rapid-Prototyping using Blender Blender proved itself to be actually not only quite useful but rather being a real treasure and workhorse for the design students.

www.ckbrd.de |- info@ckbrd.de
715 309 9795



Tangent Space Normal Maps for Blender

- by Olivier Saraja

Introduction

Normal maps have become very popular these last few years, shown as a large improvement in the Game Industry, but also in more traditional CG Industry and in the movie Industry, even if less obviously. Nowadays, solutions to sculpt models to an incredible level of detailing exist (ZBrush, Mudbox also our beloved Blender with its now famous Sculpt mode) but building normal maps to enhance the surfaces of our models is still somehow lacking. Blender now has the capability to bake Camera Space Normal Maps directly from a higher level of resolution and apply it to a lower level of resolution, thanks to UVmapping and the Multi-resolution meshes. But the very last step, the one which will bring Blender up to the standard is still missing): and that is Tangent Space Normal Maps.

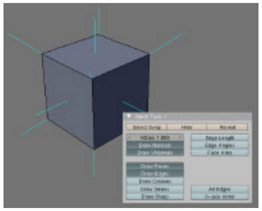
If Blender now supports these maps, it still doesn't know how to bake them

Normal maps versus Bump maps

These are different tools for different jobs. None should be considered better than the other, and sometimes, they should be combined/mixed in order to get the best results possible. They all rely on the normals of a given object, so this is the first thing we will try to address in this article) :

What's a normal?

A normal is a vector perpendicular to the surface, at the given point. In polymodeling, as complex surfaces are made of small planar faces, you can assume that there is only one true normal by face but as many normals as there are faces. This is easily viewed in Blender. For example, select your model, and then go into Edit mode [Tab]. In the Editing buttons [F9], look for the Mesh Tools 1 panel. It contain several buttons of interest, especially Draw Normals (displays normals as blue lines) and NSize (changes the displayed length of the blue line for better visualization). A normal vector has a length equal to one unit.



Bump maps

A Bump map is basically a greyscale image that will tell, for each pixel of the rendered surface, if the local normal length should be scaled. If not, the pixel will be interpreted as laying on the 'ground' level of the face (middle grey pixels of the Bump map); if yes, it will be interpreted as laying above the ground (dark pixels of the Bump map) or under the ground (light pixels of the Bump map); a 50% grey will mean no scaling of the normal. A Bump map basically simulates embossment of the surface, effecting the shading by virtually displacing the rendered pixels along the normals.

To get a Bump effect, you need to associate a greyscale texture (it could be an image or a procedural texture) to a texture channel, and activate the Nor option in the Map To panel of the Material buttons. Use the Nor slider, in the same panel, in order to modulate the intensity of the Bump effect.

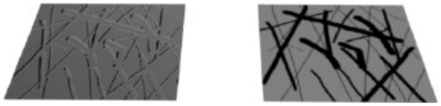
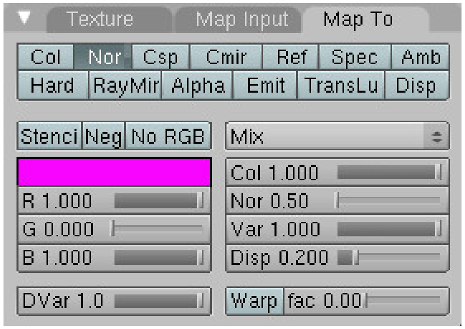


Image1: On the right, you can see the greyscale image used for the Bump, and on the left, the rendered result.

The Nor button has three states:

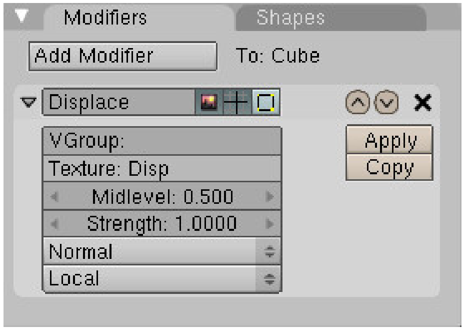
- **off** : the texture channel doesn't affect the rendered output
- **on (white)** : the texture channel does affect the rendered output
- **on (yellow)** : the texture channel does affect the rendered output but the normals are inverted (bumps become crevices)



R 1.000
G 0.000
B 1.000
DVar 1.0
Col 1.000
Nor 0.50
Var 1.000
Disp 0.200
fac 0.00

Image2: The Map To tab, showing the Nor option and its slider.

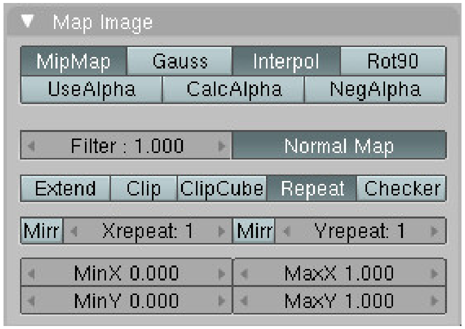
And what about Displacement maps?
Very much like the Bump map, the Displacement map will also use a greyscale image in order to actually displace the polygon along its normal according to the value stored in the texture. It is really efficient with a finely subdivided or subsurfed mesh. You can check the Displace modifier in order to learn more about this, as it is beyond the scope of the article : Bump maps and Normal maps actually fake complex surfaces on simple geometries, where Displacement maps need a very dense geometry to be effective at all.



To: Cube
VGroup:
Texture: Disp
Midlevel: 0.500
Strength: 1.0000
Normal
Local

Normal maps

In the case of the Normal maps, the normals vectors are all normalized (e.g., they have the same length equal to one unit) and a color image is used to tell, for each pixel of the rendered image, what is the simulated orientation of the local normal at the pixel. The Normal map basically simulates the local orientation of the surface by virtually rotating the pixel according to the normal. The normal effect will be quite visible even with non (or very slight) specular materials. To get a normal effect, you need to associate a color texture (an Image type one, because normal maps need special tools to be produced) to a texture channel, activate the Nor option in the Map To panel, as previously, but also to activate the Normal Map option in the Map Image panel of the Texture buttons [F6].



Filter : 1.000
Xrepeat: 1
Yrepeat: 1
MinX 0.000
MaxX 1.000
MinY 0.000
MaxY 1.000

Image3: The Map Image panel with the Normal Map button enabled.

Baking Camera Space Normal Maps within Blender

Since version 2.43, Blender has been able to bake Camera Space normals into the texture of an object. This is only a step toward a proper production of Normal Maps with Blender, because Camera Space suffers from some limitations: the normals baked are only used with the object with the same location/rotation/size versus the same location/rotation/aperture of the camera. If either the camera or the normal mapped object are moved, deformed or animated in any way, then the normals will look wrong.

I read about other normal map types, what about them?
Of course if more Normal map types exist in Blender that would be quite useful or handy : for example with an Object Space normal map animations of onlly non-deforming objects could be carried on, while with a Tangent Space normal map you can also animate deforming objects (using armatures, shape keys, softbodies, lattices, whatever) while retaining a decent normal visualization. Unfortunately Blender is not able to bake these kind of normals at the moment, so you will have to rely on external tools to build efficient Tangent Space normal maps.

Baking the Camera Space normal map into the texture of an object is pretty easy. Select the object you want to bake then UV unwrap it in UV/Image editor add a new image (menu Image » Add...) and then go in to the Scene menu [F10]. Here in the Bake tab choose the Normals option and hit the BAKE button. Don't forget to save the Normal map generated in any file format of your choice (Targa is the default, PNG tends to be very convenient).

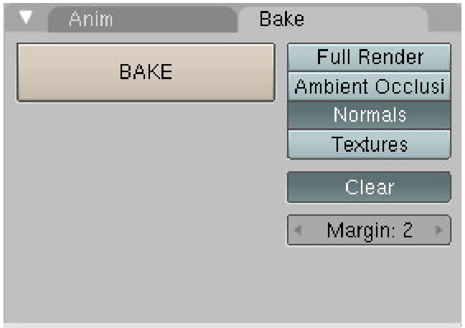
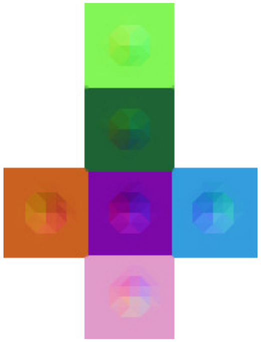
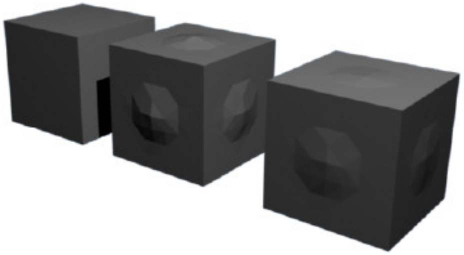


Image4: The Baking Tab.

Of course baking the normals of a mesh in order to use it with the same mesh is almost pointless. The interest is to use this baking feature with multi-resolution meshes. For example, you start with a low poly mesh that you unwrap, as specified before. In the Editing menu [F9], find the Multires panel and click the Add Multires button. Add Level and start sculpting your mesh, adding details along with levels. When you are satisfied with you sculpting, you then can check that the maximum Render level is set and proceed with the baking, just as explained before.



Example of Normal map baked within Blender from a high level multi-res of a sculpted cube



Example of render result, from left to right: normal low-poly cube, high-poly and sculpted multi -res cube, and normal low-poly cube with Normal map issued from the sculpted high-poly cube

In the previous image, the Normal map has been computed in the Camera Space for the central cube, but applied to a low-poly instance of the same cube, slightly on its right. Of course, at first glance, the fact that the normals are wrong (they are not exactly in the same Camera Space) is not easy to spot in this still shot, but within an animation, wrong shadows will quickly betray the wrong Normal map.

Testing this kind of Normal map within an animation should be a good illustration of how much Camera Space could be useless when used on objects that actually move or are relocated in a different location than the one the Normal map has been built from. Unfortunately, as stressed before, the Camera Space Normal map is the only one that Blender actually knows how to build. Object Space Normal maps would be more suited to any animated rigid model. Finally, only a true Tangent Space Normal map would be useful at all with any animated deforming object. The last part of this tutorial is about how to produce such a Tangent Space Normal map, with external tools, and how to use it within Blender.

Using DNormGen for building Tangent Space Normal Maps with Blender

There are many free tools available on the web for building Tangent Space Normal maps, but only a few are available for many OS at the same time. One of these, actually stands out as having a quite good support for Blender, is the Drag[en]gine Normal Map Generator and is available for Windows, Linux and Os X. Source codes are also available under GPL v2.0. Go here.

Go to the Downloads section and select the package you want to use. There are dependencies to solve for Unices and Os X, but nothing too fancy (libfox1.4, libfox1.4-dev, libpng3, libtiff4, libxmu-dev are a few examples) and perhaps link the libtiff.so.3 to libtiff.so.4 (sudo ln -s libtiff.so.4 libtiff.so.3).

Once DNormGen has been extracted, pay attention to the dnormgen executable and the dragengine_dim_export.py file in the scripts/ directory. There is also a lot of good information on the usage of the tools in the README file.

The Blender script that is shipped with the tool will take two models : the low-poly one should be named *.low (in the OB: field) and the high-poly one should be named *.hi (* being the original name of the object). This is absolutely mandatory for the script to work, and could be done in the Link and Materials panel of the Editing menu [F9] buttons. It will then export the two models in the .dim format, which is specific to the normal generator tool. Then this tool will be able to build a Normal map from this .dim file.

1st step : From Blender

There are quite a few steps to do within Blender, before using the script, like renaming the objects, setting Subsurf on the high-poly one, and selecting the two objects to export. The exact procedure is as follow :

1 Select your low-poly object and add to its name, in the OB: field, the extension .low. Unwrap it, give it a material (if it doesn't already have one) and a texture (Image › New... in the UV/Image editor) that you need to save (Image › Save or Image › Save As...).
2 Now select your high poly object and add to its name, in the OB: field, the extension .hi.
3 Add a Subsurf modifier to the high-poly model (a Level 0, as specified in the documentation, is no longer possible in Blender whose minimum is 1; if you don't want to alter the shape, use Simple Subdiv. option instead of Catmull-Clark).
4 Select the two objects : the low-poly one and the high-poly one.
5 Change one of the views into a Text Editor. Use File › Open... in order to load the dragengine_dim_export.py script. Hit [Alt]+[P] with the mouse cursor pointer in the Text Editor to run the script, choose a name for the exported .dim file (or accept the default one).

That's it for the part to be done within Blender. You should now have one .dim file that you can feed the DNormGen tool.

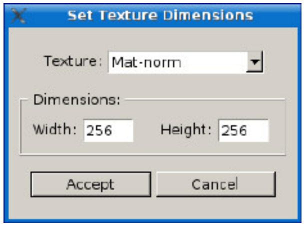
2nd step : From DNormGen

When you run the DNormGen tool, you will get two distinct windows :

1 **DE Normal Map Generator :** this window will show the 3D models (low-poly, high-poly) and will let you configure the kind of map you need (Displacement map, Normal map, Tangent Space or Object Space, etc.)
2 **Texture Preview :** this window will show the resulting maps.

At first, you will need to load the previously created .dim file in the DE Normal Map Generator (File » Open...). The original low-poly should appear on the screen. Using the Left Mouse button, you can click and drag the 3D object to make it rotate around its center. You can also zoom in or out by holding [Shift] down and using, the very same way, the Left Mouse button. Now you can check and explore the View menu. You can decide to show the Low-poly model or the High-poly model. If the maps have already been computed, you can also show them in this window.

The next step, for the generation of a Tangent Space Normal map lies in the Texture-Maps menu: you will have first to define the Texture Maps Size, by setting the Height and Width. These parameters defaults to 256.

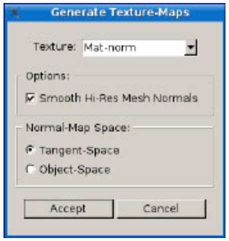

Set Texture Dimensions

Texture: Mat-norm

Dimensions:

Width: 256 Height: 256

Now, you can proceed with the generation of the Normal map. In the Texture-Maps menu, select Generate Normal/Displacement Maps and pay a few seconds attention to the pop-up options window.


Generate Texture-Maps

Texture: Mat-norm

Options:

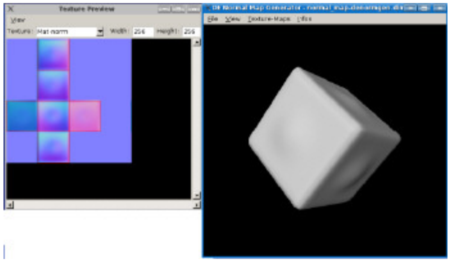
☑ Smooth Hi-Res Mesh Normals

Normal-Map Space:

● Tangent-Space
○ Object-Space

At first you need to specify if the High-poly should be smoothed before proceeding (check box Smooth Hi-Res Mesh Normals) then, you can select the kind of Normal Space map you need (radio button Tangent-Space or Object-Space).

After accepting the options the texture map is generated and the Texture Preview window automatically updated. You will note that it shows the UV coordinates set on the low-poly object and that the newly computed Normal map fits the proper space. From the Texture-Maps menu Save Normal Map in the location of your choice. It should default to a normal.tga name.


Texture: Mat-norm Width: 256 Height: 256

I have some troubles, please help!

The borders of my Normal map are slightly messed up? In View › Set Hi-Res Mesh Options, increase very slightly the Vertex Offset option. Alternatively, using a greater map could help, too. The bumpiness given by the Normal map is not strong enough? In Texture-Maps menu, select Apply Detail Normal Map and experiment with the Strength value. Applying it with the default Strength of 50 could lead to dramatic results, so use carefully. I read somewhere that the Green layer of my Normal map should be inverted. Is that true? It used to be true, yes, but Blender's normal mapping now is fully compatible since v2.43 (previously the scale of the Z vector differed). So you don't have to invert the Green channel in a software like The Gimp anymore in order to use DNormGen normal maps within Blender if you are using the latest version.

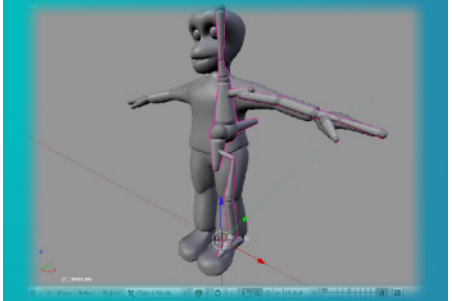
3rd step : Back to Blender

You now have a correct Tangent Space Normal map to be used in Blender. Select the Low-poly model you unwrapped a few minutes ago. In UV Face Select mode, in the UV/Image editor, discard any temporary map you might have and load the normal.tga built by DNormGen using the menu Image › Open... In the Material buttons, menu Shading [F5], check the Shaders tab and activate the NMap TS button.

In the Map Input tab, the texture should use the UV coordinates, and in the Map To tab, it should be set to Nor channel (invert it if you feel the shading is wrong). Finally, in the Texture buttons, in the Image Map panel, check that the Normal Map button is activated and that the map selected in the Image panel is the normal.tga built by DNormGen. Save [F2] your project, that's it, you are done with a Tangent Space Normal mapped low-poly model!

- Olivier Saraja

AutoRig for Blender



Great news for character animators especially those who use blender for modeling and animating characters for computer games.

Blenderhead Niels Krook has just released a script titled AutoRig. The script automatically rigs up the character and all you have to do is model an character change few properties and fire up auto rig from the scripts menu and your character is ready for animation.

The AutoRig script can be downloaded from: http://niels.degooier.net/blender_scripts/

AutoRig also come with installation instructions as well as complete documentation on how to use it.

Niels website has more scripts for animators go check it out.

Creating Realistic Flame in Blender

- *by Krzysztof Zwolinski*

Introduction

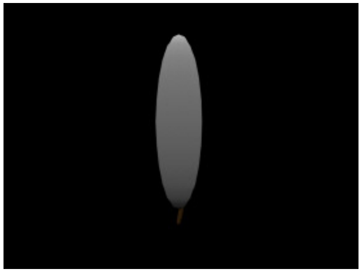
'KeepTheFlame' is one of my works inspired by the book of Pete Drapera, "Deconstructing the Elements with 3ds max 6”. Even though this book is aimed at users of 3ds max, I recommend this book everyone.

How to make flame?

Before we start to make materials we have to take a minute and wonder about characterization of our object. In this tutorial we make a flame from a lighter, lets think about it?:

- Flame is fading out with the environment
- It has two different colors, half blue and yellow
- Each half has a different property..

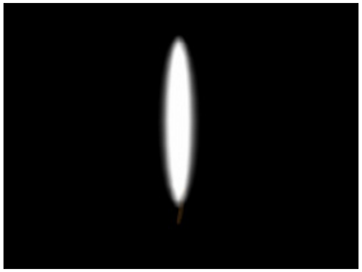
Now it seems that we can't make this flame just from a simple material. So to do this material, we will use Node materials. In this tutorial we use a simple object Sphere scaled in Z. The Flame in the 'KeepTheFlame' was also made in this way, but it takes a little time and effort to set up everything to have this effect felt right.



Lets start:

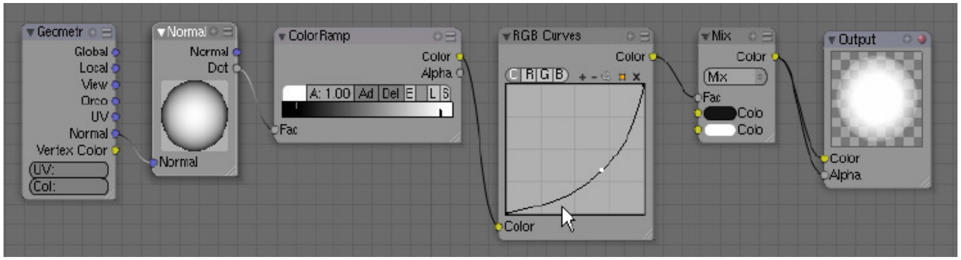
First we make a simple material and Turn on ZTransp , Alpha 0,0 , Spec 0,0. Now for the Node, in Node editor we erase everything leaving only *Output* (I am not going to explain about the properties of every Node we have in the editor, I will only show you how to connect them and what this connection results in). Now add *Output*, *Geometry*, *Normal*, *Color Ramp*, and RGB Curves nodes. Mix and connect them like on the image below.

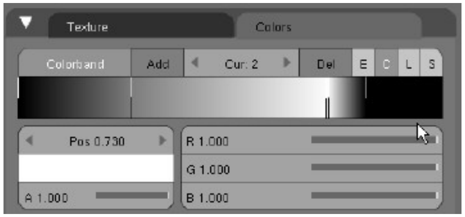
We use a setup like this almost all through this tutorial as it gives great adaptability. In this case the object is covered with two colors which are setup in Mix Node, and Mix Node is setup by ColorRamp and Normal Node. So we can setup fading one color to another on the object Normals. Later we change Normal Node to Texture Node.



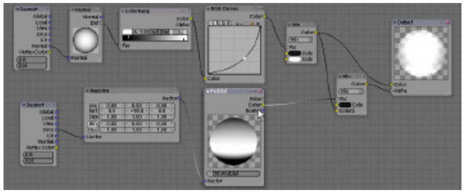
And this is the result of this setup. Gentle fading of the white color in the black and because we also put this setup in the alpha channel the object disappears and fades out with the background.

Now we have an object that is gently fading, so now we have to split him into two different parts. First we make Blend texture like this(Image on Next page.).

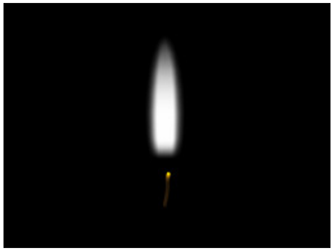



Texture | Colors
Colorband | Add | Cur: 2 | Del | E | C | L | S
Pos 0.730
R 1.000
G 1.000
B 1.000
A 1.000

Copy all of the Nodes and change Normal, ColorRamp on Texture, Mapping and set up adequate coordinates of the texture,

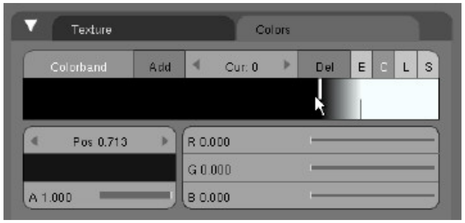


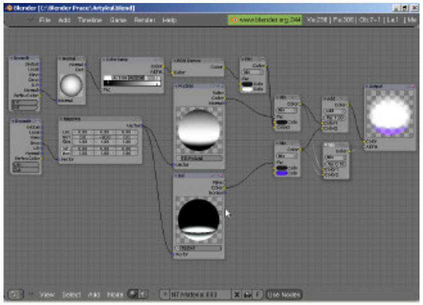
and the results.



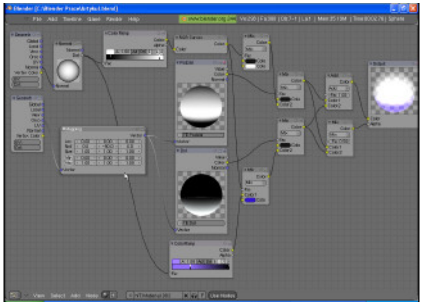
Our flame is now split by the texture in two colors, white and the black. In the alpha channel, black is transparent. So now we put all of the Nodes made in first step to fadeout what is in the white slot of Mix Node.

Create another Blend texture, which will replace the Black slot in the Mix Node and add another Mix Node and mix them together like this.

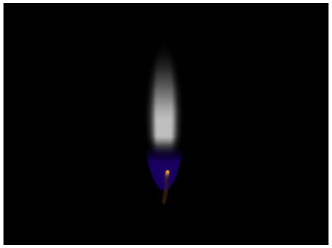

Texture | Colors
Colorband | Add | Cur: 0 | Del | E | C | L | S
Pos 0.713
R 0.000
G 0.000
B 0.000
A 1.000



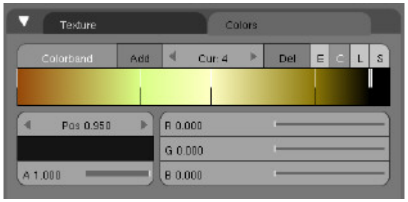
If we render it now we can see that the blue part of the flame is not fading out, that's because the Mix from Normal fade in is not attached into the blue slot. So now copy them and set them up like in the image.



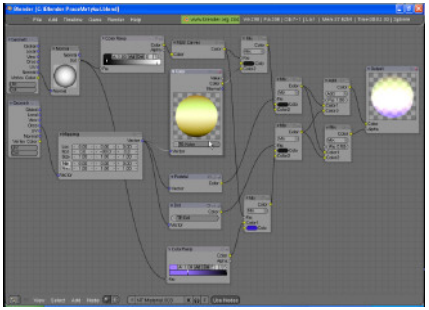
Also copy Normal and ColorRamp and put them into the black slot, this give us the ability to change the outer line of the blue part. If you setup everything correctly you should be have some thing like this.



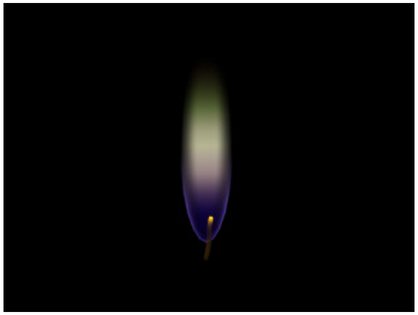
Now we can take care of the upper part of the flame. Make another Blend texture like on this image.


Texture | Colors
Colorband | Add | Cur: 4 | Del | E | C | L | S
Pos 0.950
R 0.000
G 0.000
B 0.000
A 1.000

And connect this texture to the first Mix Node we created to the white slot



Now we have complete setting for the flame.

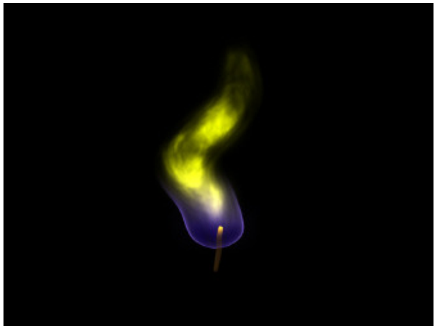


Now its up to us and what we want to achieve. We could add a Material Node with Emitter and cloud texture. With emitter we could use Radiosity and the alpha channel in the clouds to make the flame more real.

Conclusion

You can see that making a good Node material is hard work and you need to study them in real life. Most of time the best results are made by trials and error.

Information in this tutorial are only the basics of the Node Editor and making of the flame materials, but I hope that the basics and study of the image of the final settings of Nodes used in the work 'Keep The Flame' will let you to make more realistic flames and many more Node materials in future.

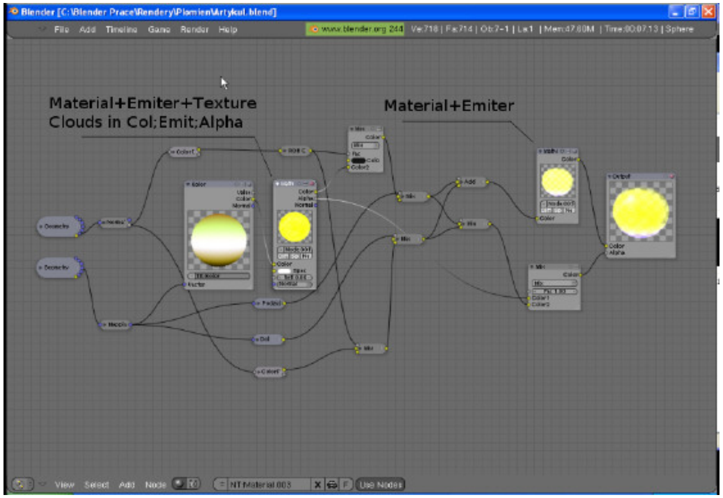


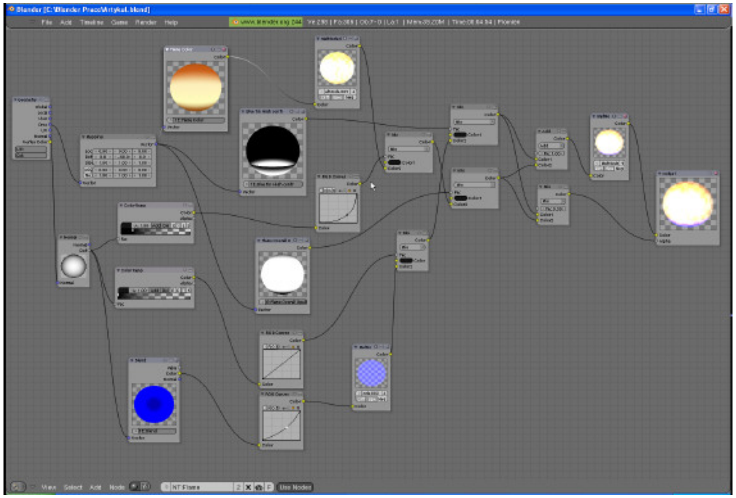
Krzysztof Zwolinski
(aka Blender-Man)
Poland

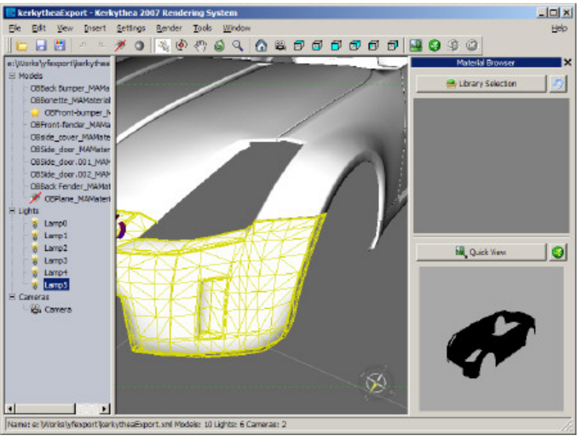
I have work with 3D softwares for almost 10 years. I started out on AutoCAD then moved to 3dsmax. Throughout this time I worked on many programs but found Blender to be the best of them.

I currently study journalism with specialization on "CyberMedia and Electronical Communication". In my free time I work as 3D graphic freelancer.

k.zwolinski@wp.pl







Blender to Kerkythea

- by Abhishek Gupta

Introduction

Since the development of Yafray has slowed down a little, I was frantically searching for some other external renderer which could be used with Blender. So for the past few months I have been trying my hands on renderers like PovRay, SunFlow, Indigo etc. Then just a month ago I bumped into Kerkythea which does have a bit support for blender too. But why Kerkythea?

- It has a good development pace
- It's the most feature rich renderer for Blender.
- Has six different global illumination methods
- A very easy to use G.U.I. (this is the best part).
- A superb physically accurate material editor.
- Layered material support
- Download-able materials
- Has a python exporter from blender
- Has a Blender build with Kerkythea integrated just like Yafray.
- Supports windows and Linux platform

That's really great. But still it has got some glitches which can be easily sorted. Here are few to name

- Its integration with blender is not on par unlike Yafray.
- Its separate build and exporter are still in W.I.P. So if you want a renderer that is tightly integrated just like Yafray, than Kerkythea is not for you.
- Currently Kerkythea is good for stills only.

But even after all these cons, the result I got from my initial renders are so good that I just couldn't ignore it. So here is my tutorial so that everyone can try Kerkythea. This tutorial is going to be about a basic global illuminated render. So hopefully every one should be able to follow it.

Requirement for this tutorial

Kerkythea 2007.
Blender to kerkythea build.
dlls for the blender build.
That's it.

PART 1: Modeling and setting up the scene

I have chosen a very simple scene which I have re-created from the tutorial for Sunflow renderer (another very good renderer) which appeared few months ago in Blender magazine. First fire up the Blender to Kerkythea build. Create the simple scene as shown in the figure (fig 1). Assign basic diffuse colour to all the spheres. Place a light in the scene and select lamp button. In the Kerkythea shadows and photons panel, click on enable and shadow button. Also go to the world button select black colour as the world colour. I will be explaining later on why black world is chosen.

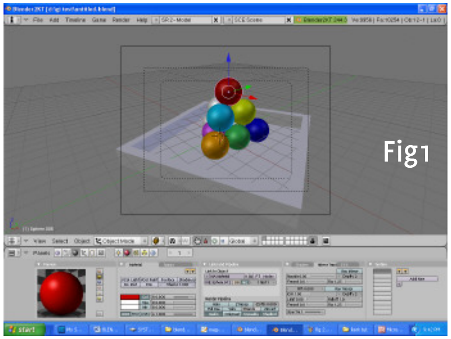


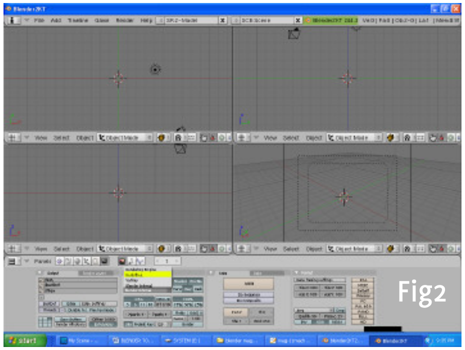
Fig1

PART 2. Exporting

Now press F10 and go to render panel. Here you will see a drop down menu just below the big render button. Click it and select kerkythea from it (fig 2). After clicking it you will see two more tabs will appear in the render panels. Select kerkythea from it. Move your mouse towards the 3d windows and click on the display type (fig 3). Select user preference. Click on the file path button. Go to the YFexport panel and assign a directory of your choice. Now come back to the render panel. In the kerkythea tab deselect Exec. Kerkythea button. Hit the render button.

What the hell just happened?

You just exported a file in a format which will be read and understood by Kerkythea. Why I have gone for such a painful procedure? because all the settings like material editing etc. will be done in Kerkythea itself.

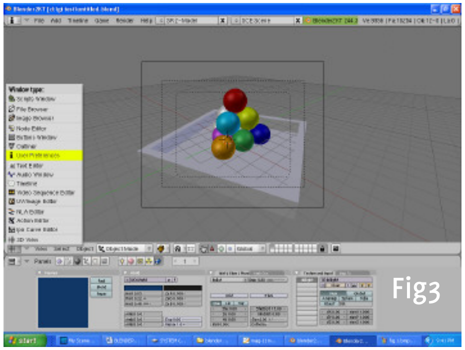


PART 3. Importing

Here comes the fun part. Fire up Kerkythea 2007. (This is the latest at the time of writing this.). I will be really brief about what you are seeing right now. The Kerkythea is divided into about four panels. On the left side is the tree for all the objects in the scene. In the middle is the 3d view. On the right there is a material library selector and below it a quick preview panel with a small green render button.

This is going to come in handy. On the top is the usual file open, save etc. menu. The rest I will describe as we move along.

Click on File››Open or just press [Ctrl+O], navigate to the location where you have saved your export from blender and hit OK.

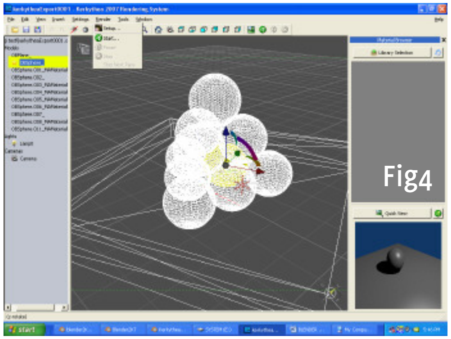


PART 4: scene setup

Welcome to Kerkythea! After loading the scene you should notice two major changes. On the left pane you should see all the objects from the scene. In the middle you should see a wireframe view of the scene.

Move on to the right side and click the render button (yes the green one). Congrats on your 1st render. Now for the magic. In the left pane double click on the name of one of the spheres. A star should appear in the left indicating its selected (See fig 4). Now right click on it and select edit material from the pop up menu. Now we are in the material editor button.

Material editor has lots of tricks up its sleeve so I will leave that for you to explore. For now just reduce the specular by clicking and dragging on the dial towards left side under colour tab. Press apply and close editor. Do this with all the objects.



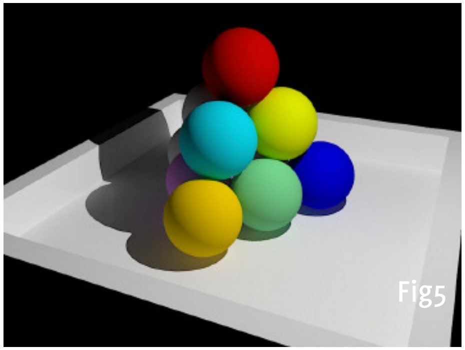
PART 5: Global illumination

Remember I asked you to make the sky colour black? I did it on purpose. To understand that, here is a little insight on G.I. (global illumination). GI is a lighting technique which takes into account not only direct light from light source present (like the lamp in our scene) it also takes the sky colour and light that bounces off the objects. Since I didn't wanted the objects to receive light from the sky I made it black. You will understand it more as we proceed.

In the top panel select render and then setup. In the ray tracer tab click on the AA method dropdown menu and select production AA. In the soft shadows menu select medium. Now click the global illumination tab. Kerkythea has lots of choices for a global illumination method. In the method panel select Photon Mapping + Final Gathering (SW). Set the no. of photons to many. In the final gathering menu select very many under rays tab. Set the accuracy to very good. Set the gathering depth to 2 and press OK.

Now click on the render button. Sit back and relax as this is going to take lots of time depending upon your system speed.

After the render is complete you can view the result and save it by pressing window in the top panel and selecting rendered image. Now look at the subtle bleeding on the floor and on the edges of all the spheres. Look at the shadows which are not that dark thanks to global illumination (see figure 5). Isn't it beautiful? This is the best result I have ever seen in an external renderer for blender.



Conclusion

Kerkythea is a very advance renderer and support a lot more than you just saw. In the future I am looking forward to the following changes:

- 1.A very good integration with blender so that a lot more parameters can be tweaked from within blender.
- 2.Hopefully with the new upcoming UI changes in the blender, a lot more tweaking will be available for external renderers like kerkythea.
- 3.The GPU accelerated global illumination is made available.

Links

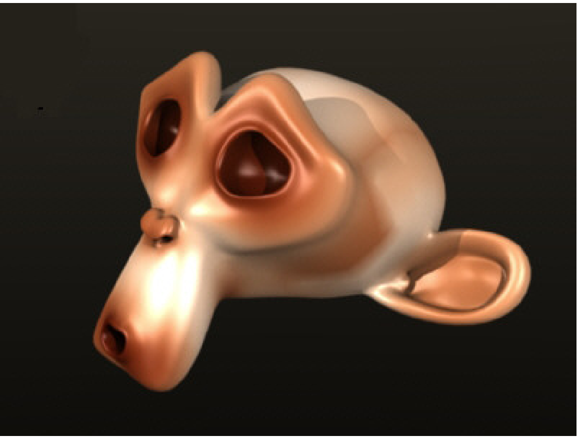
Newer Versions of Blender2Kerkythea can be found here.
Kerkythea can be found here.

Till then happy blending and rendering.

Abhishek Gupta
India

I am currently pursuing C.A. (chartered accountant) which does take my most of the time. But thanks to Blender it has given me the power to pursue my dreams.

abhifx@yahoo.com



Texture Seam Removal Using Multi-UV & Render Bake

- by Eric Pranausk

Introduction

Triangles, artifacts and seams... all perfectly acceptable in the real world, but commonly despised in computer graphics.

Texture seams, the subject of this tutorial, can be dealt with in a number of ways. Often, they are strategically placed in some discrete location: clothing borders, hairlines, under a belt. Textures or materials can be blended to hide the seams. The method discussed here involves the complete removal of seams using Blender's Multi-UV layer feature in conjunction with Render Baking.

Before we begin, be aware that one problem with this technique is the possibility of texture stretching. With the difference between two different UV Mapping layouts, pixel details can become distorted. Two ways to combat this is to use high resolution maps and careful UV unwrapping (unlike the 'quick & dirty' examples featured here!).

PART 1: The First Map

Start by selecting and marking the edges you want to use as UV seams. Notice the bright orange edges I've marked as seams, on the modified Monkey mesh shown in Figure 1, along with the result of the UV unwrap to it's right.

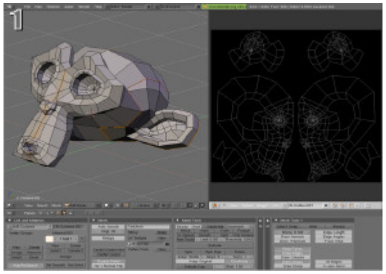
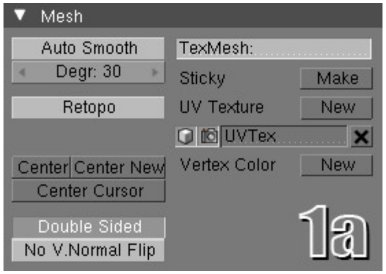


Figure 1a shows the Mesh panel with the default UV set, named "UVTex". At this point, you may want to give it a more meaningful name (I've used "UV1" in this example).



Mesh

Auto Smooth

Degr: 30

Retopo

Center | Center New

Center Cursor

Double Sided

No V.Normal Flip

TexMesh:

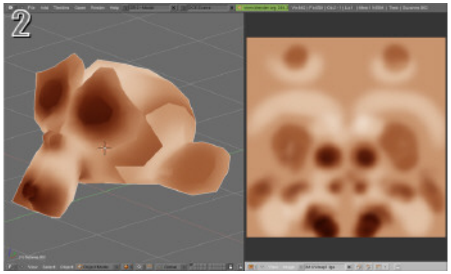
Sticky | Make

UV Texture | New

UVTex

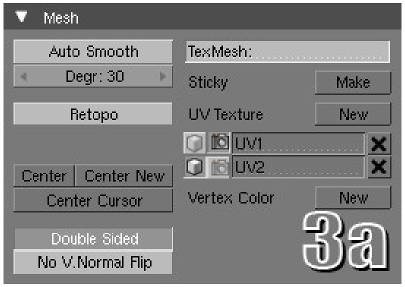
Vertex Color | New

Create a new image (or, open an existing file) in the UV/Image Editor, and set the 3D view to Textured to see the map on your model. After saving the UV layout, I used it as a guide for painting a quick texture in an image editor. The map and textured mesh (with very visible seams) can be seen in Figure 2.



PART 2: The Second Map

Now we create the second map where the seams will be fixed. Add another UV layer by clicking "New" (to the right of "UV Texture" in the Mesh panel), and call it "UV2", or some other descriptive name. Notice in Figure 3a that icons next to the UV layer names indicate which UV set is active in the editor (cube), and for rendering (camera).



Mesh

Auto Smooth

Degr: 30

Retopo

Center | Center New

Center Cursor

Double Sided

No V.Normal Flip

TexMesh:

Sticky | Make

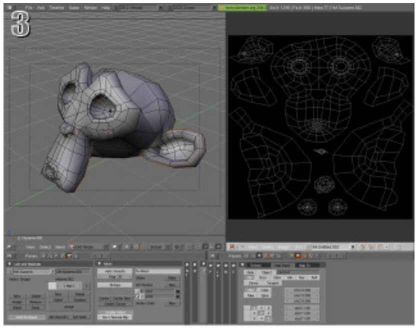
UV Texture | New

UV1

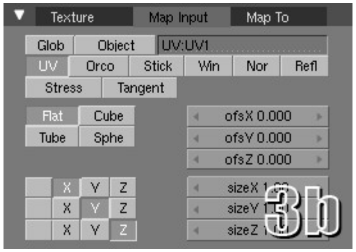
UV2

Vertex Color | New

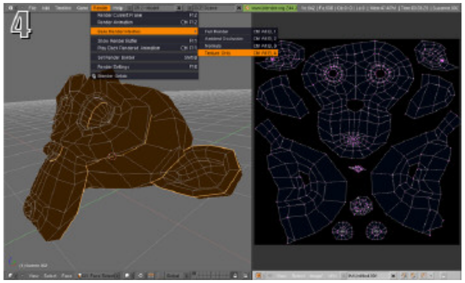
To create a new UV layout for this layer, clear the original seams and mark different edges as new seams. These seams can cross the old ones, but should not share any edges. My new seams and the results of a new UV unwrap can be seen in Figure 3.

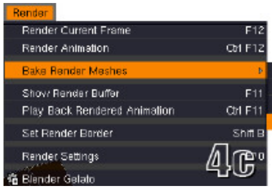


Before the second map can be baked, our mesh must have a material applied, with the first map in the color channel. Also, be sure that the correct UV set is used for this map. Figure 3b shows the Map Input tab of our material, with the texture set to use UV coordinates, and the "UV1" layer.


Texture | Map Input | Map To
Glob | Object | UV:UV1
UV | Orco | Stick | Win | Nor | Refl
Stress | Tangent
Flat | Cube
Tube | Sphe
ofsX 0.000
ofsY 0.000
ofsZ 0.000
X Y Z | sizeX 1.00
X Y Z | sizeY 1
X Y Z | sizeZ 1

With the mesh selected and in UV Edit mode, the second map can now be baked. Once the baking process is started, the blank image in the UV Editor window (Figure 4, right side) will soon be filled with the pixels of the original map but in a different layout. Figure 4a shows the menu path and hotkeys for the Render Baking functions. For seam removal, use the "Texture Only" option.




Render
Render Current Frame F12
Render Animation Ctrl F12
Bake Render Meshes
Show Render Buffer F11
Play Back Rendered Animation Ctrl F11
Set Render Border Shift B
Render Settings
Blender Gelato

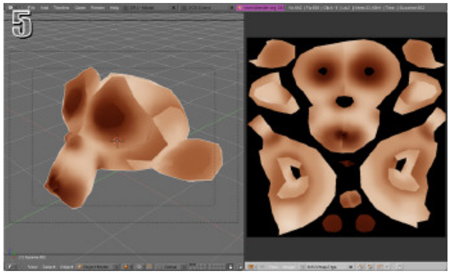
Full Render Ctrl Alt B, 1
Ambient Occlusion Ctrl Alt B, 2
Normals Ctrl Alt B, 3
Texture Only Ctrl Alt B, 4

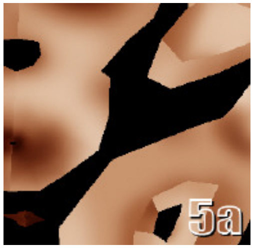
NOTE: Blender 2.44 seems to have a Render Baking bug, so 2.43 or a later version may be used for this step.

PART 3: Seam Removal

Once the Render Bake has finished, you will have a new map as seen in the UV/Image Editor in Figure 5, and the seams are still visible on the mesh in the 3D View. But Figure 5a shows that they are no longer on the edges of our UV layout. Being located more central to the individual UV islands, they can now be removed with your image editor of choice.

Figure 6 shows the results of some quick modifications to the Render-Baked map on the SubSurfed model. The location of seams has not been completely disguised in this example, but the hard edges have been 'smoothed' to show that those parts of the map can now be edited without obvious divisions in the map.





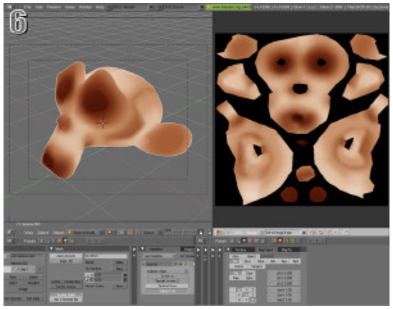


Figure 6a focuses on the softened details of the first map's seams within the second map's layout. Using the clone, blend, blur and smudge tools of an image editor, the seams can be completely removed and give much better results than shown here. And remember, to avoid introducing any new seams into the second map, do not allow your editing to touch the edges of the UV islands. You can use selection and masking tools to keep your changes within predefined areas.

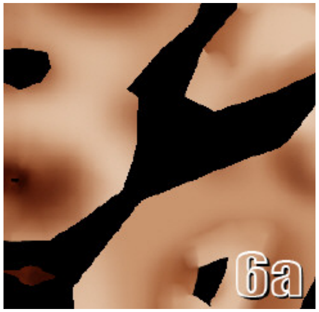
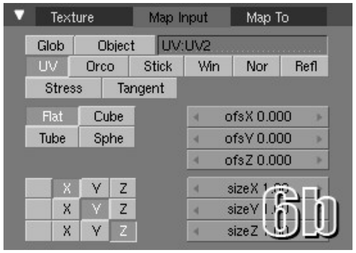


Figure 6b shows that we are now using the UV2 coordinates in rendering, which is needed for our seamless map (a modified version the second map) to render correctly. Be sure to save the fixed texture, and load it into the color channel of the material. Now you're ready to render!



UV:UV2

ofsX 0.000

ofsY 0.000

ofsZ 0.000

sizeX 1.00

sizeY

sizeZ

Figure 7 shows the SubSurfed mesh with the first map applied. Notice the unsightly seams.

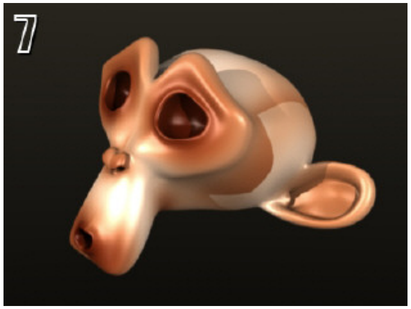
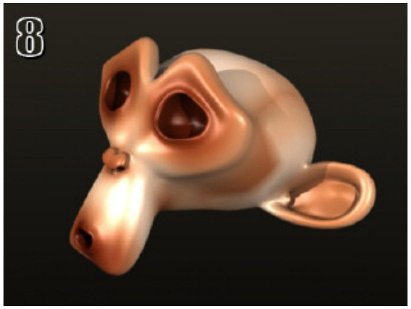
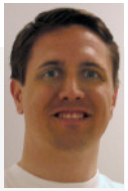


Figure 8 shows the SubSurfed mesh with the final map applied. The original seams have been subdued, and could have been completely removed . . . it just takes a little more time than I was able to invest into this tutorial!



NOTE: This tutorial was made with intermediate to advanced users in mind . . . hence the lack of detailed descriptions or hotkey references. For beginners and any who require additional information or explanation of terms, please utilize the search functions and other resources at the websites listed below.



Eric Pranausk
(aka Cire)
USA

I'm a graphic designer for a manufacturing company in California, the husband of a beautiful woman named Carolina, and user of a program called Blender.

ericncaro@gmail.com

Links

http://www.blender.org/tutorials-help
http://wiki.blender.org/index.php/Main_Page
http://blenderartists.org/forum
http://www.blendernation.com
http://forums.cgsociety.org/forumdisplay.php?f=91
http://www.google.com/search?hl=en&q=blender+3d

Blender News

Great things are on the blender horizon!

Blender 2.45 Released!

Blender 2.45 has been released. This is a bug release version and a large number of bugs and fixes have been taken care of to create a stable release for the next round of improvements. Go pick up your copy in the normal places.

October 1st will see Project Peach officially started, although a great deal of work has already been done. Concept art is being finalized as well as work on the script as a result of a pre-production workshop attended by our team. There has been a change in the team members as Lyubomir had to drop out for personal reasons and was replaced by William Reynish (one of the team alternates.)

Arno Kroner, trainer at Disney Feature Animation, teaching at Woodbury University and organizer of Siggraph 's FJORG agreed to come to Amsterdam to organize our first week. He's planning to do a full week workshop about all aspects of visual storytelling, storyboarding, layout, scene planning and acting. He's also helping us now as script consultant.

Main Blender feature work is of course the hair rendering and editing. Both topics should be tackled by mid November latest...

We will need tree/plant generating software, and probably landscape editing stuff. Here Python might be a great tool.
The Blender Conference is set to kick off in Amsterdam October 12-14 at the newly opened multi-media center De Zwijger, located in the Amsterdam cultural docklands, on the river IJ.

During the Blender conference we'll reveal the movie concept and basic character sketches.

Pre-orders for the Peach Project DVD are being taken through the e-shop. Money from the pre-sales will help fund the project as well as individual donations which can be made through the Project Peach site.
http://peach.blender.org/index.php/about

Donations of 30 Euro or more: you can get your name mentioned in the movie credits, like for the DVD pre-sale credits. Please tell us which name.

Donations of 250 Euro or more: you can get your name mentioned as Main Sponsor in the movie credits.

While you are at the e-shop be sure to check out the available books for sale. Not only will it increase your blender knowledge, but it helps support the Blender Foundation. The e-shop currently has three excellent titles in stock:

Essential Blender

Introducing Character Animation

French Blender Guide

Making of “Roof Tops”

- by Alex

Introduction

This render started out with a simple idea that I had: to create a scene of an old dormer. Soon, I was photographing textures, trying out lighting, and modeling more objects for the rest of the scene. Unlike most art, this piece wasn’t planned; it just sort of expanded from the dormer out. In this article I’ll explain the components of the final image, focusing on texturing.

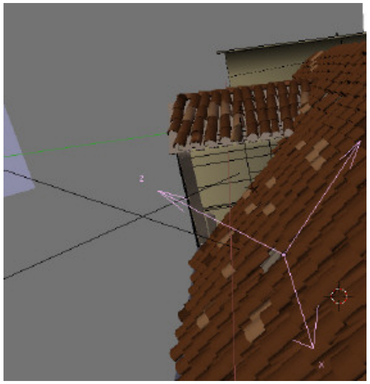
Tiles

The most common question I was often asked about the render is, “How did you do the tiles?” The modeling needed the most thought, but here I won’t go into detail on that. You can find my tutorial on modeling those tiles at: blenderartists.

Reading that tutorial will give you some information on the modeling. I recommend that you read it before continuing, so you'll have a better understanding of the whole process. Once the array modifier on the tiles had been applied, I assigned a simple material, called "rooftile" to all the tiles. Having a red-orange colour, this would be the main colour of the roof.

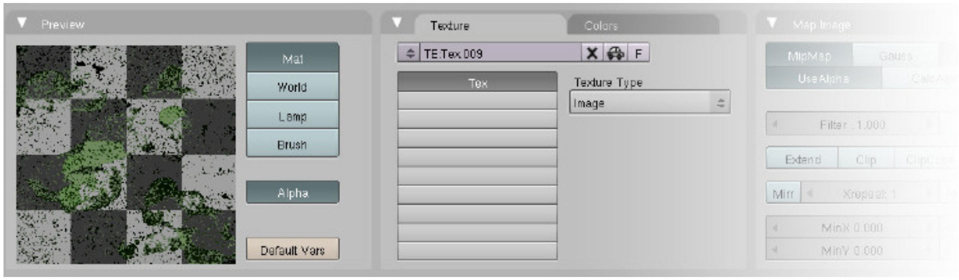
I then created two variations of the material: one darker and one lighter. I assigned these to random tiles (having separated the tiles into individual objects). This gave a more random, old-looking effect (as if the tiles were made of different types of clay). The next step was to use the ‘randomize loc rot size’ script to slightly jumble the tiles. Then I joined them back into one object.

This re-joining is an important step (although it’s not ideal for adjustments of the tiles’ positions later on) because it allowed me to have one moss texture spread out across all the tiles. I used an empty to determine the position of the texture.



Painting the moss texture itself turned out to be a simple process in Photoshop. I added noise to a plain green image, erased sections, used the clone tool to make adjustments, added a bit more noise – you get the idea.

The texture was then applied to the tiles, affecting Col and Nor. The use of the texture as a bump map was essential, because it prevented the moss/dirt from looking completely flat. However, you’ll notice it doesn’t look particularly raised either; creating that type of thick moss would require a different technique.



TE:Tex.009

Texture Type: Image

Filter : 1.000

Xrepeat: 1

MinX 0.000

MinY 0.000

Note: Joining the tiles created one object with three materials. I knew that I would have to apply the moss texture to each material to have moss on all the tiles, but I chose not to. Most of the tiles are of the main colour (material: "rooftile"), with only a few lighter or darker ones in the mix, so it created an effect of varied cleanliness across the tiles.

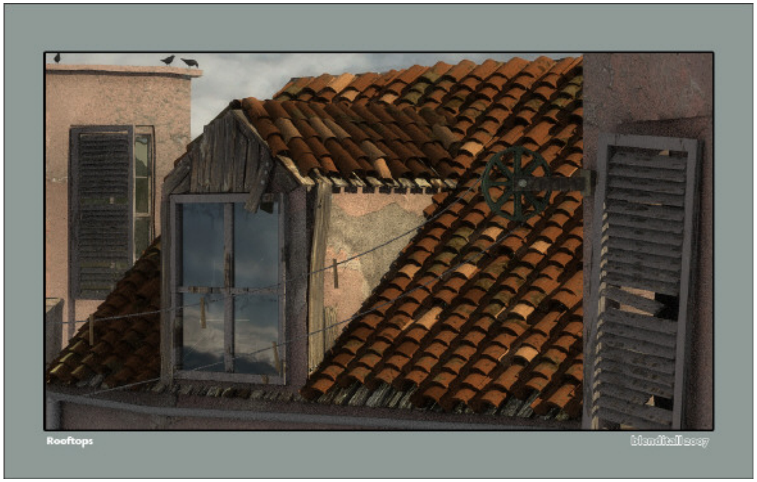
Wall Textures and Effects

Texturing the walls didn't prove to be as difficult as I had expected. I simply went on a sort of texture hunt around town, and ended up photographing a local running track (you may notice it's not in great shape) for the wall texture. After a bit of tweaking in Photoshop, I applied the texture to the dormer wall using an empty (the same method used for the moss). On the dormer, the texture tiled unpleasantly around the object, but that didn't matter since a window and planks of wood are covering most of the rest of the surface.

The textures for the other walls were taken from the same place, but there's nothing usual about how they were applied. The final step for the walls, to add a touch of realism, was to add multires to the meshes and sculpt some chips out of the edges of the walls (noticeable on the dormer, on the edge closest to the camera). Simply beveling didn't look as good.

Wooden Planks

One of the challenges of texturing is to prevent repeating images, especially on separate objects that all have the same texture. To avoid this problem on the wooden planks, I created one material for all of them with an old wood texture (also required hunting). Then I joined some of the planks into one object, causing the texture to be stretched across those planks. The planks that I didn't join with any others kept the original texture. As long as there are only a few objects that remain on their own (and thus each have an identical texture placement), this creates a look of variation – and only one image texture is used.

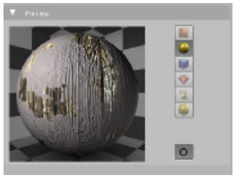


Shutters and Window frame

The material for the shutters and window frame consists of (from top to bottom) a wood texture (the same one used for the planks), a cloud texture, and the same wood texture again.

- The first wood texture affects Nor only.
- The cloud texture acts as a stencil, and also affects Nor.
- The third texture, wood again, affects Col and Nor.

This whole setup gives the effect that rough old wood was painted, and then the paint was chipped away over years. Note in the below image that even where the wood isn't showing, you can see the grain, as if the wood is underneath. This happens because of the first wood texture being used as a bump map. Also, the Col slider of the first texture can be adjusted to mimic paint translucency. Try a similar setup yourself, it's fairly easy and very useful.



Conclusion

That's all of the most important texturing information about this render, and I hope you found it helpful. I find that the best way to come up with original textures is to just get out there and take pictures of them. That way you're guaranteed that: a) The image is 100% yours, and b) it's unique.

BlenderArtists Forum thread here:.

Alex (aka blenditall) blendergallery

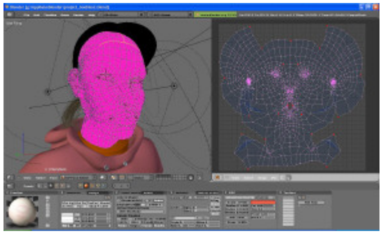
Making of "Skin Shader"

- by Schuh

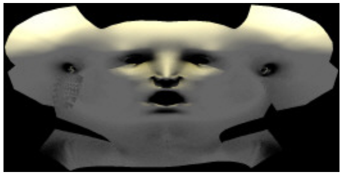
Introduction

A lot of people asked me for a little tutorial about my skin shader in this picture, and although it's quite simple it might not be easy to understand for someone who isn't into the nodes and SSS features of blender yet.

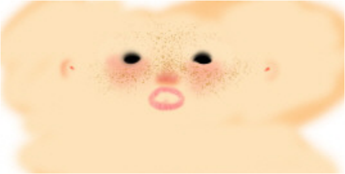
So here's my approach:
I began, of course with modeling the head and unwrapping it to create the necessary textures for the shader(btw my unwrap is really crappy so don't do this at home kids ^^).



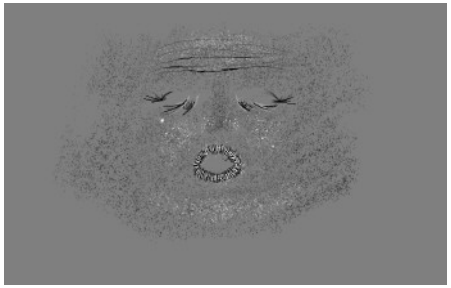
I then created new image of size 2048x1024 which is kind of small for that kind of texture, but it was enough for me now. I used blenders rather new "Bake Render" function to create a base layer so I can orientate myself with it in gimp. This was my starting point for all the other layers.



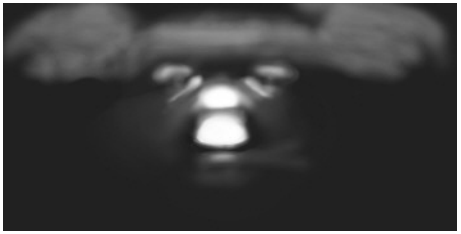
On one layer I painted the basic colors without shading plus some extras, like freckles, makeup and such things. Note that this layer only contains colors without any shading because all the shading will be done by the renderer. Basically I keep the whole image a very light skintone except the lips, the nose and other areas that generally appear more reddish because of thinner skin or higher blood vessel concentration.



Next up was the bump map. I started with a 50% grey layer and painted black strokes and dots for anything that goes inward and white lines for everything outward. This is how I created pores and wrinkles and the basic structure of the lips. A few white dots here and there make the skin look kinda unclean and more natural. Note that different skin regions have different sized pores and so on, just look in the mirror to find out :).



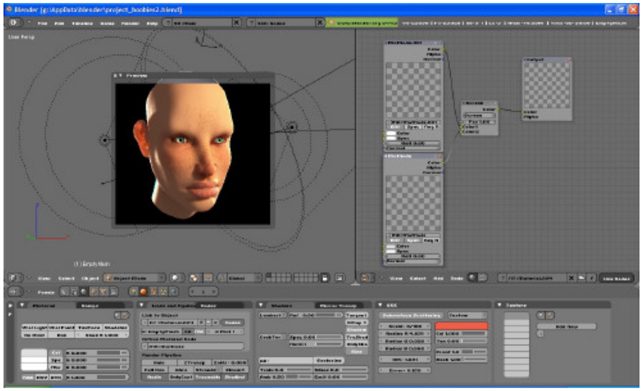
Last texture I did was the Specularmap, which basically defines what areas are shiny and not. Depending on what kind of face you're doing and what happened to the face there are different areas to paint... normally it's just the more oily areas like the nose that are shiny but we also need the lips to be shiny so I painted them a bright white. I also added a slight shine on the forehead and the cheeks. For this texture I didn't start with a 50% gray because this would mean 50% shiny everywhere which would be way too much so I started with a very dark grey. I also kept the transitions between the areas smooth.

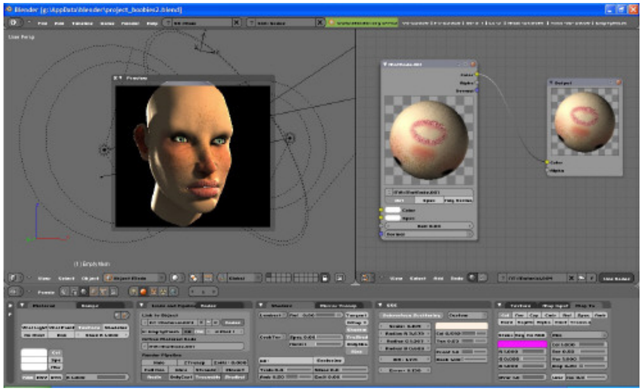


So now we have all our textures and are able to start on designing the material. I began by creating a new Nodes material and adding a simple diffuse matnode, which in normal blender way got the diffuse map set to "Col" and the bumpmap set to "-Nor".

Then I activated Subsurface Scattering for this matnode and set it to a very slight typical skin setting for a very subtle effect. With the "tex" slider at 0.5 it scatters the texture a little bit and for us simulates the epidermal scatter.

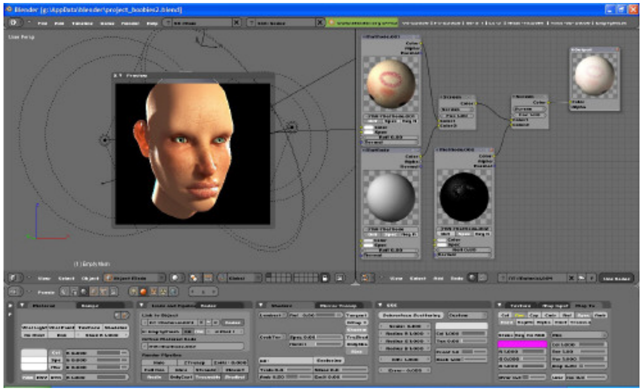
Now on to the subdermal scatter. I added another node, without textures this time, set to a heavy scatter in a dark red tone, which was then screened over the epidermal node. You need to play around with the settings a lot to get the right effect for your picture. Remember: sometimes less is more. The older or the masculine the person is the less scattering is typically present.



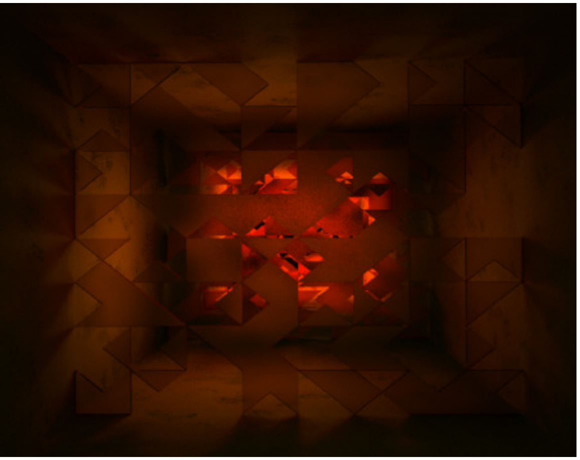


Now onto the shine. So why didn't I just activate the specular settings in the other matnodes?

Because blender blurs the speculars together with the diffuse color. Sometimes this effect might be interesting but in my case I wanted to have the speculars crisp. So I added another matnode set to specular only, which was again screened over the other two nodes. In the specular node I had 3 textures: the specularmap set to "Spec", the bumpmap set to "-Nor", "Spec" and "Hard" (this way there's less specular in the dents and more shine on the Zits :), and again the specularmap set to "Hard" but with the DVar slider set to 0.2. This way I got around creating an extra gloss map, because in this case the shiniest spots are also the glossiest. The Dvar slider is used to tune the effect down a bit so the gloss is not too hard.



Hope this helped :)
The rest is basically
just good lighting.
- **schuh**.



Making of "Tangrams of Light"

- by Adam Friesen

Introduction

I decided to write this article after receiving an invitation from Blenderart magazine to write an article about the lighting of this piece. To start with the basics, the modeling was done with a bunch of extruded meshes with triangle faces randomly deleted and the materials used were a glossy material for the glass and a cracked material for the hall.

Now on to the lighting

At first I thought, I'll do the usual, try to place lamps to give it an atmospheric effect, but quickly found it was harder than it looks. To try to achieve a good atmospheric effect with proper balance of light and dark is no easy task. When Broken developed soft shadows for all lamps except hemi, the job got a little easier, but was still no easy feat. I wanted decent contrast within the image as well.

At first, I just popped together some lights of many different colors with shadow lamps for shadows, but as you can see I scrapped that idea and went with a largely single color theme for the lights and very weak Ambient Occlusion. I knew bad lighting could really break the image. I wanted the mood to be dark, but not overly gloomy, but also to have an impression of hope that says it's not all bad, and there's light ahead.

The bright orange-red lights in the back

These were to provide an effect of looking at the bright lights through the glossy surfaces. To give decent contrast, I cranked up the power of the lamps so the gradient from light to dark is visible easily enough without any contrast adjustments in post processing. In both images, lights placements are usually behind several layers and can give the glossy surfaces a kind of glow, giving a better atmospheric effect. In the first image, I used nine of these lamps to provide evenness and thus not cause any unwanted darkening in the corners. I also set the lamps to sphere, the reason I do this is because I can get an idea of just how far any light from those lamps would go. Previous setup attempts included mixing the high powered lights through different areas of the corridor, but the contrast achieved was poor in spots and I didn't wanted that.

The shadow only lamp

Shadows can really help improve the atmospherics of the scene and provide a better relationship between light and dark as well as bring depth in the scene. Because of the high filter value of the gloss materials, some shadows came out colored as well, which was a wanted effect. Broken's build helped a lot, because it enabled me to use the soft shadows, which look much better aesthetically than sharp shadows. To many sharp shadows everywhere was not an appealing options, as the soft effect treatment for the scene would be shattered. In both images these lamps were usually between the bright lights behind several layers and the camera, they're also usually placed off center, generally at this step lots of test renders were done to see if the position of shadows were good. I also noticed they also made some areas almost black but figured that they helped the mood

The yellowish lights

The two types of lamp I just described can make things look pretty, but that wasn't enough. I knew I could've used a light just in front of the camera to help illuminate some of the closer glossy surfaces a little bit better. This light was mainly for this task, and was placed a little towards the corner for the reason that in the first image, the lighting would've looked too symmetrical otherwise. The second image had the camera even farther from the bright lights in the back, so for this reason I needed a way to illuminate some of the closer surfaces. That light isn't as powerful and used a different color to prevent contrast problems, but also of a harmonious color to match the color scheme, these lamps also cast shadows.

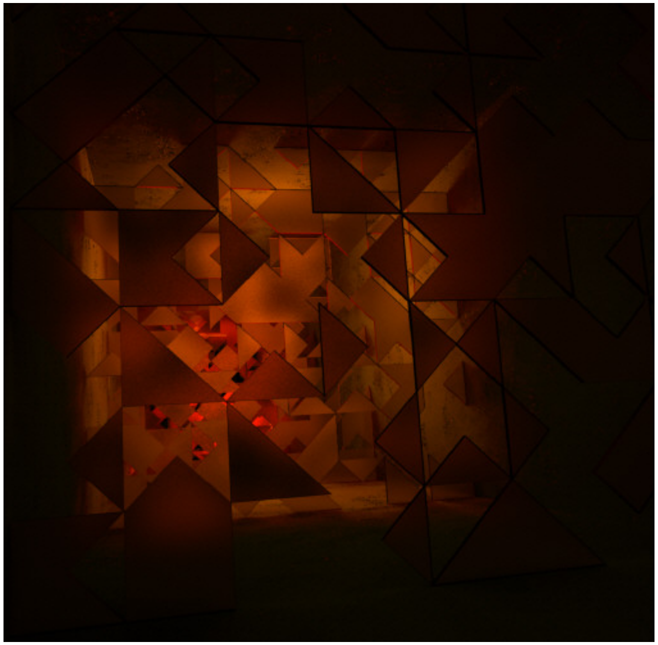
The negative lights

The problem I had with the scene is that it appeared brighter near the camera, and because of that it would be little difficult to draw the viewers eye down the hall, and trust me I've made images where the lighting looked weak. The general goal for using negative lamps was to be darker near the camera rather than down the hall. These lamps were placed near the camera area and made the area almost black giving a reason to look down the hall, and offers smooth increases in brightness farther down, sometimes darkening a little but that's okay.

The effect is more easily seen in the second image, where you can see that the front surface is almost in complete darkness and then you see the transition between light and dark. Sometimes negative lights are important in drawing the viewers eye. Here, you don't want the viewer looking at the edges or the front, but draw him towards the brighter area. As you notice in the second image, the first surface in the way is darkly illuminated to give an impression of it being in front of the brighter surfaces behind it and to draw the eye towards the brighter area. Noticing the surface layer, I know I've done this on purpose as part of providing the mood.

Summary

So it took me hours of lighting setup to get a good relation with color, light, and dark. Almost any material can be enhanced with the right lighting. Lighting to draw the eye, lighting to provide contrast, or lighting to provide a seemingly mysterious glow in an area to attract the viewer. Lighting with shadows to help provide the mood, as a great deal of moodiness is set by the lights. In this scene bright lights can bring about a generic place mood, but here you see that the lights and the darkness purposely left and brought in, sets a sort of mysterious mood that intrigues the viewer, materials are enhanced like as in glossy transparent materials where a light from behind can give them a surreal glow or SSS materials can use lights to give it a glow in an area that contributes to the scene rather than take away from it.



Lighting can help achieve what your vision for the image is, which is maybe to scare, intrigue, or interest viewers, I also purposely left the AO to be very weak as I needed dark space in the latter ideas of how to light it as I wanted this to be sort of dark but not overly gloomy and have areas of light where your eye doesn't have to remain in darkness, or if you're in that dark hall then you may find there's hope to be found in the red area. But take heed that in scenes like this, atmospheric lighting can be tricky and it may take a number of lighting setups and many test renders to get it right.

This image was ripe for atmospheric lighting and for many abstract type pieces, finding and placing the right lights with the right shadows could be the difference between making it good or being slammed into oblivion with comments about bad lighting.

Hope you Enjoyed. Happy blending!

Adam Friesen
USA

Adam Friesen is a graduate of Maize High School in Maize, Kansas. He is presently involved in 3D art production and is an active member of the Blender community.

ajfriesen7@cox.net

Here is how!

1. We accept the following:

- Tutorials explaining new Blender features, 3dconcepts, techniques or articles based on current theme of the magazine.
- Reports on useful Blender events throughout the world.
- Cartoons related to blender world.

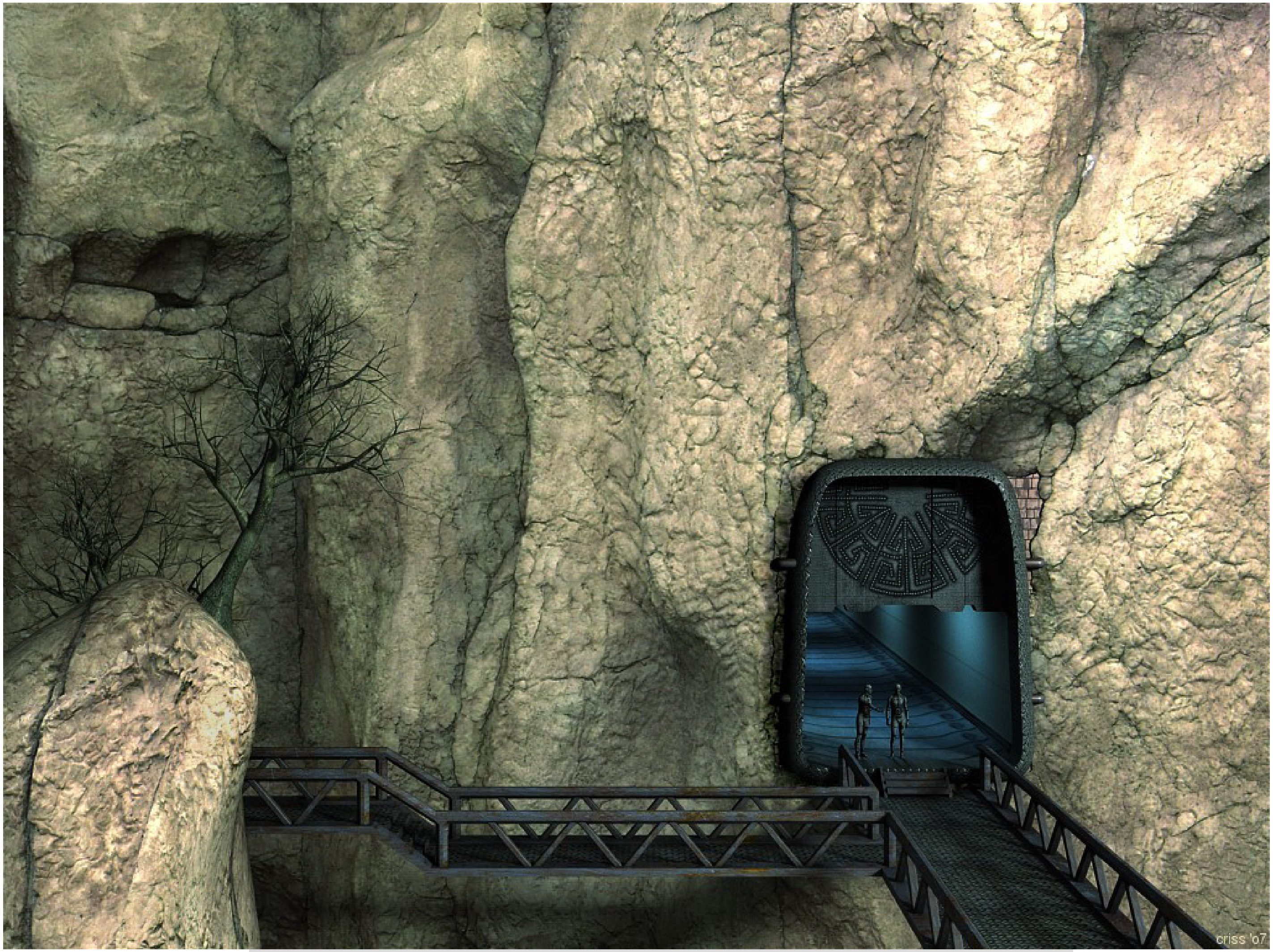
2. Send submissions to sandra@blenderart.org. Send us a notification on what you want to write and we can follow up from there. (Some guidelines you must follow)

- Images are prefered in PNG but good quality JPG can also do. Images should be seperate from the text document.
- Make sure that screenshots are clear and readable and the renders should be at least 800px, but not more than 1600px at maximum.
- Sequential naming of images like, image 001.png... etc.
- Text should be in either ODT, DOC, TXT or HTML.
- Archive them using 7zip or RAR or less preferably zip.

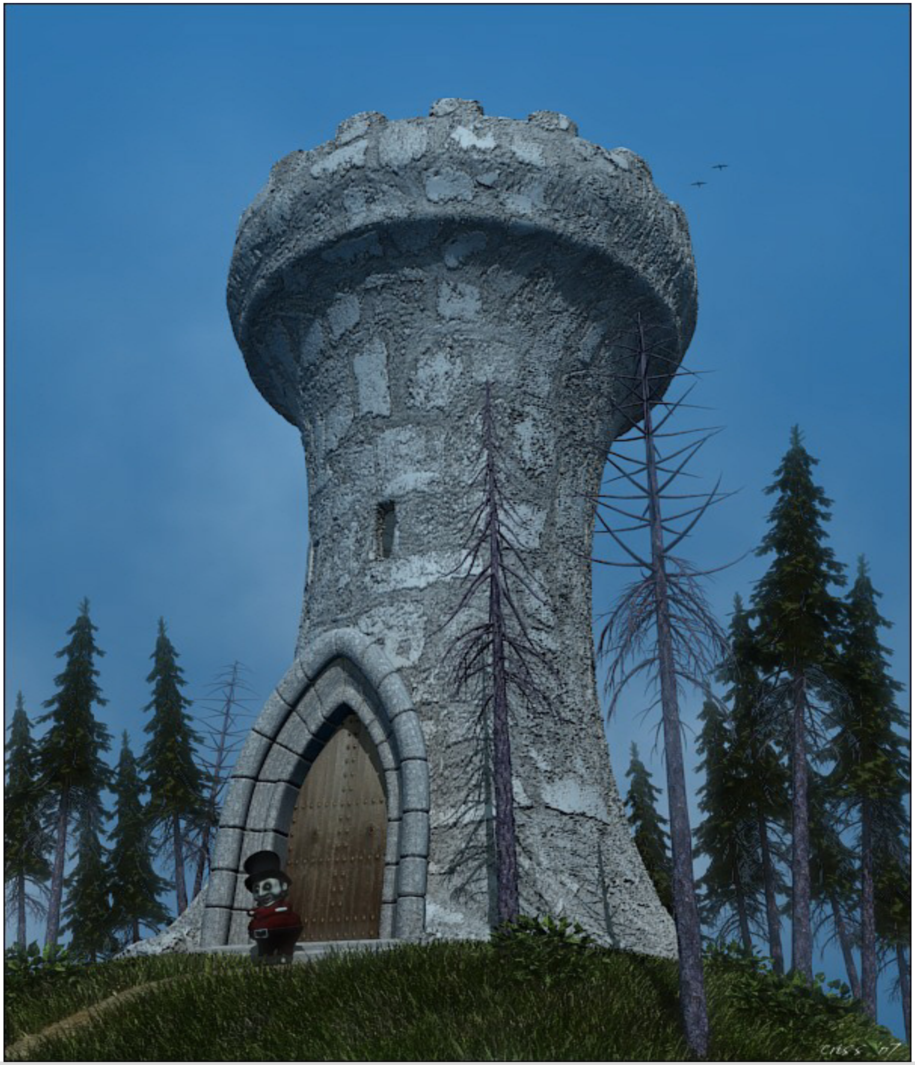
3. Please include the following in your email:

- Name: This can be your fullname or blenderartist avtar.
- Photograph: As PNG and maximum width of 256Px. (Only if submitting the article for the first time)
- About yourself: Max 25 w ords .
- Website: (optional)

Note: All the approved submissions can be placed in the final issue or subsequent issue if deemed fit. All submissions will be cropped/modified if necessary. For more details see the blenderart website.



Cristian Mihaescu - **Asthope**



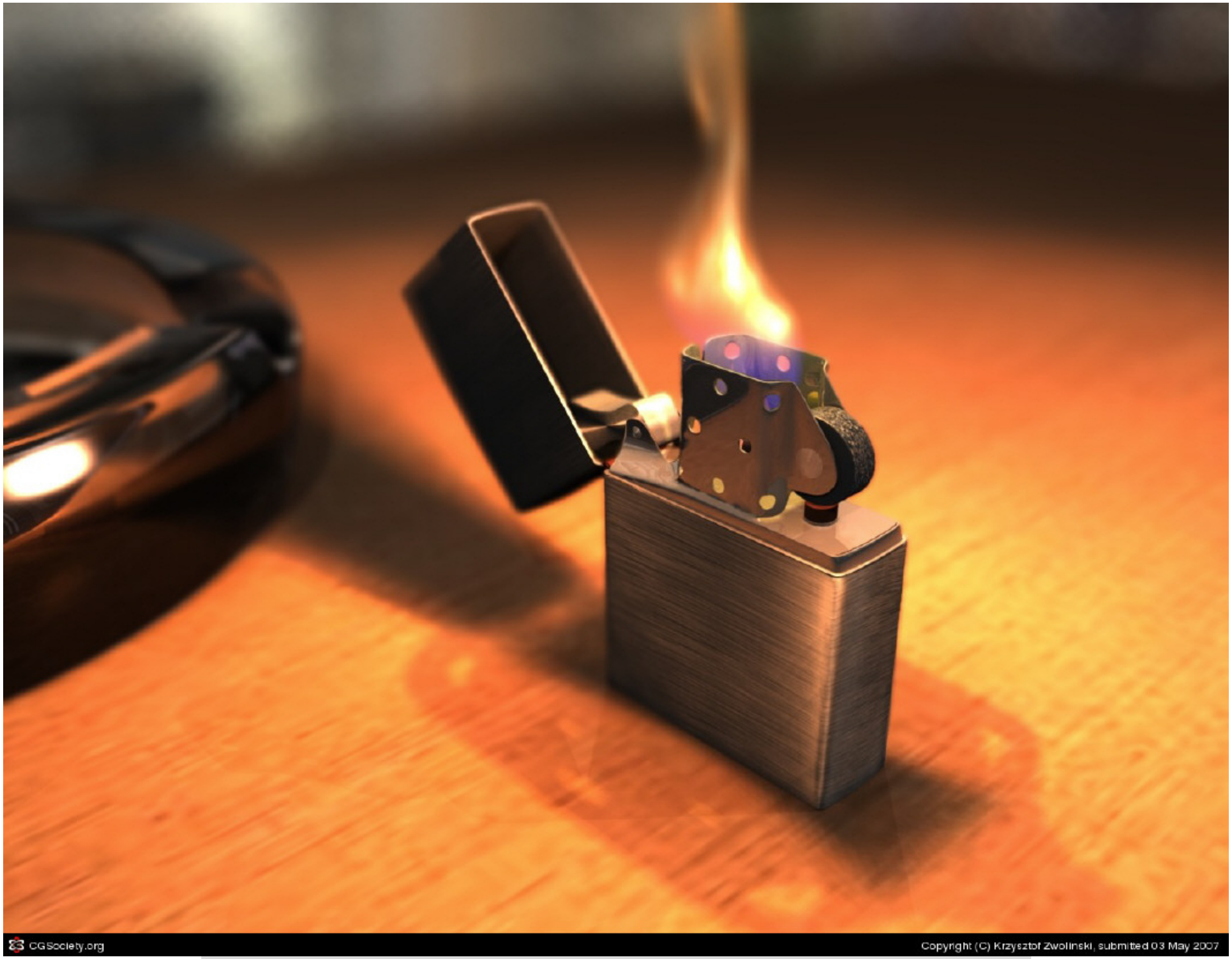
Cristian Mihaescu - **Owner**

Al-Haitham Jassar © 2007 Jassarbrush.com

Al-Haitham Jassar - **The Croc**

Copyright © G. Max, 2007. URL: http://www.georgemax.co.nr/

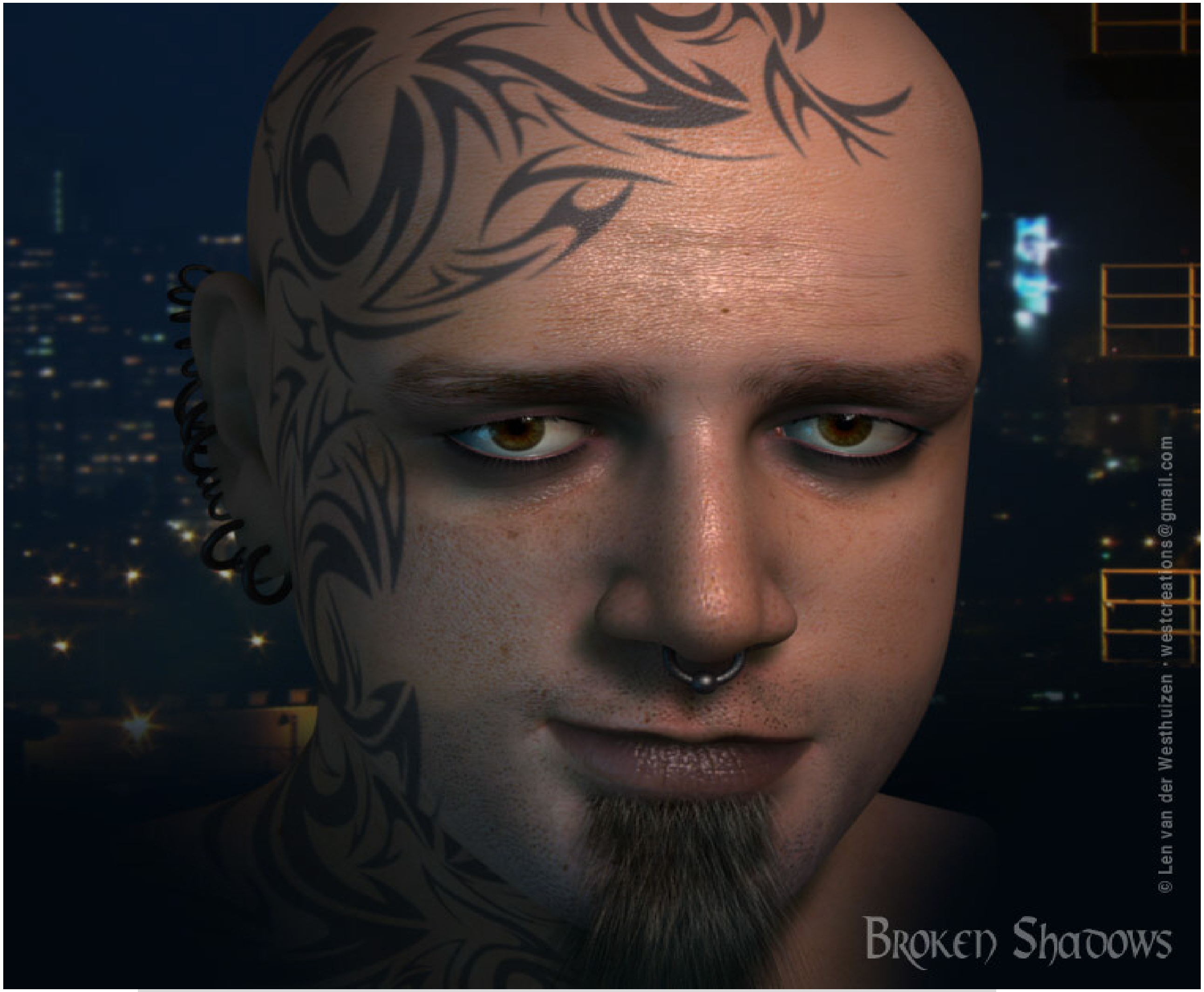
George Max - **Lemonade**



CGSociety.org

Copyright (C) Krzysztof Zwolinski, submitted 03 May 2007

Krzysztof Zwolinski - **KeepTheFlame**

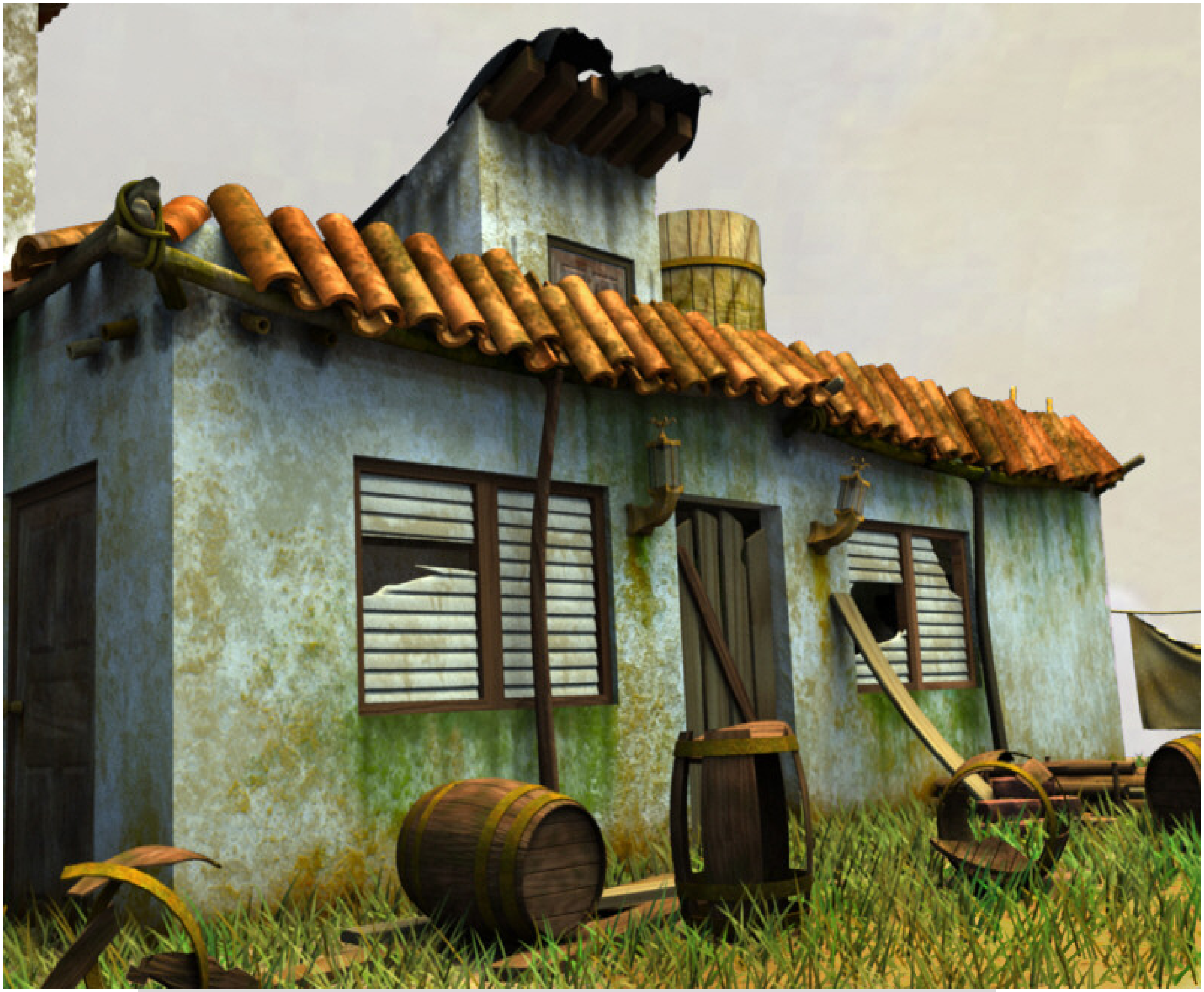


Len van der Westhuizen - **Broken Shadows**

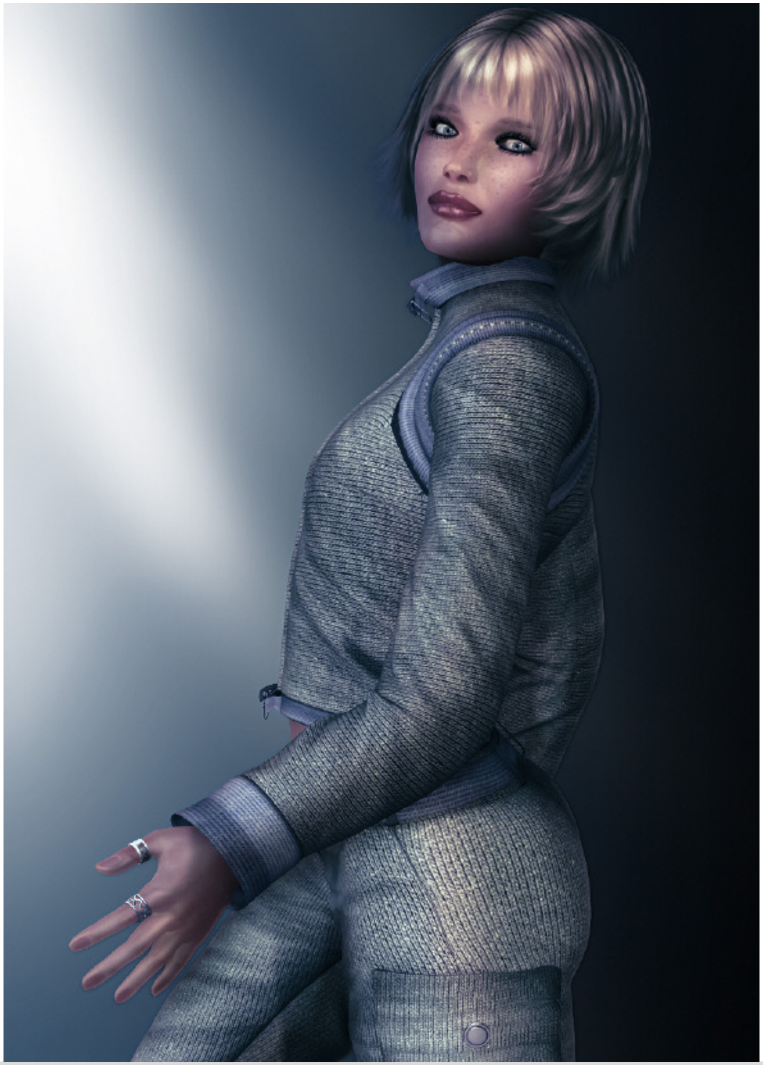
MASTER DANIX
STOP
URBAN SCENE
MARCOS PIRES

Marcos Pires - **Urban Scene**

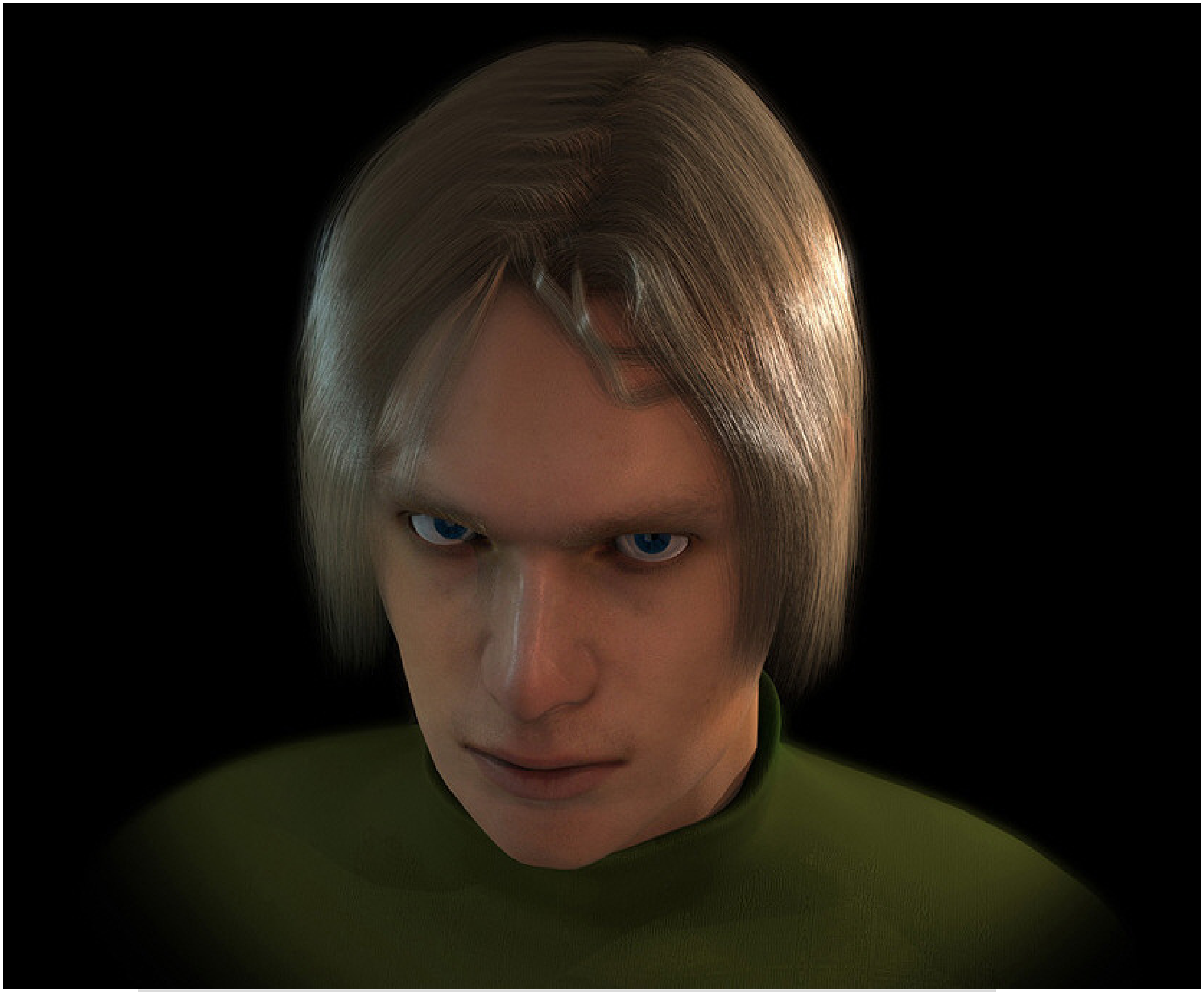
Rafael Tenorio-elGordon - **Padre Lagarto**



Yain Rodrigo Vieyra - **Casita**



Zoltan Miklosi - **Dangerous Jane**



Petri Rantanen - Self Portrait

Issue#12 Sep 2007

Theme: Fantasy

- Fairies
- Dragons
- Fantastic Creatures/Landscape/Plantlife Any other flight of fantasy found only in the Artists mind

Disclaimer

blenderart.org does not takes any responsibility both expressed or implied for the material and its nature or accuracy of the information which is published in this PDF magazine. All the materials presented in this PDF magazine have been produced with the expressed permission of their respective authors/owners. blenderart.org and the contributors disclaim all warranties, expressed or implied, including, but not limited to implied warranties of merchantability or fitness for a particular purpose. All images and materials present in this document are printed/re-printed with expressed permission from the authors/owners.

This PDF magazine is archived and available from the blenderart.org website. The blenderart magazine is made available under Creative Commons' Attribution-NoDerivs 2.5' license.